

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
«Ғ.ДАУКЕЕВ АТЫНДАҒЫ АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС
УНИВЕРСИТЕТІ»

Коммерциялық емес акционерлік қоғамы
Телекоммуникациялық желілер және жүйелер кафедрасы
«ҚОРҒАУҒА ЖІБЕРІЛДІ»
Кафедра меңгерушісі _____ Темырканова Э.К.
(ғылыми дәрежесі, атағы, Т.А.Ж.)
_____ « ____ » _____ 2020ж.
(қолы)

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: Дейкстр алгоритмін зерттеу және оны программалау тілінде модельдеу

Мамандығы 5В071900 Радиотехника, электроника және телекоммуникациялар
Орындаған Жансай Қыдырәлі Қалманұлы Тобы РЭТк(МТС)-16-2
(Т.А.Ж.)

Ғылыми жетекшісі т. ғ. к., доцент Хизирова Мухаббат Абдисаттаровна
(ғылыми дәрежесі, атағы, Т.А.Ж.)

Кеңесшілер:

экономикалық бөлім бойынша:

доцент Тузелбаев Бакберген Ибадиллаевич
(ғылыми дәрежесі, атағы, Т.А.Ж.)

_____ « ____ » _____ 2020ж.
(қолы)

өміртіршілігі қауіпсіздігі бойынша:

доцент Жандаулетова Фарида Рустембековна
(ғылыми дәрежесі, атағы, Т.А.Ж.)

_____ « ____ » _____ 2020ж.
(қолы)

есептеу техникасын қолдану бойынша:

т.ғ.к доцент Хизирова Мухаббат Абдисаттаровна
(ғылыми дәрежесі, атағы, Т.А.Ж.)

_____ « ____ » _____ 2020ж.
(қолы)

Нормобақылаушы: аға оқытушы Накисбекова Балауса Рыскожаевна

(ғылыми дәрежесі, атағы, Т.А.Ж.)
_____ « ____ » _____ 2020ж.
(қолы)

Пікір беруші: _____

(ғылыми дәрежесі, атағы, Т.А.Ж.)
_____ « ____ » _____ 2020ж.
(қолы)

Алматы 2020

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
«АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ»
коммерциялық емес акционерлік қоғамы

Ғарыштық инженерия және телекоммуникация институты
Телекоммуникациялық желілер және жүйелер кафедрасы
Мамандығы 5B071900 – Радиотехника, электроника және
телекоммуникациялар

Дипломдық жобаны орындауға берілген

ТАПСЫРМА

Студент Жансай Қыдырәлі Қалманұлы
(Т.А.Ж.)
Жобаның тақырыбы Дейкстр алгоритмін зерттеу және оны программалау
тілінде модельдеу

2019__ ж. «_11_» _11_ №_147_ университет бұйрығымен бекітілді.

Аяқталған жобаны тапсыру мерзімі «_____» _____ 2020__ ж.
Жобаға алғашқы деректер (талап етілетін зерттеу (жоба) нәтижелерінің
параметрлері және зерттеу нысанының алғашқы деректері): _____
Графтың алты төбесі бар олар (1, 2, 3, 4,5,6)
1-ден 2-ге дейінгі қысқа жол - 7, 3 - 9, 4 - 20, 5 - 20, 6 - 11.

Диплом жобасындағы әзірленуі тиіс мәселелер тізімі немесе диплом
жобасының қысқаша мазмұны: _____

Кіріспе

1 Желіні суреттеудің, талдаудың және синтездеудің математикалық
аппараты

2 Қысқа ара қашықтықты табу әдістері

3 Дейкстр алгоритмін Delphi программалық тілінде моделдеу

4 Өміртіршілік қауіпсіздік бөлімі

5 Техникалық-экономикалық бөлімі

Қорытынды

Әдебиеттер тізімі

Графикалық материалдардың (міндетті түрде дайындалатын сызбаларды
көрсету) тізімі: _____

Желі құрылымының графы;

Берілген желі үшін байланыс матрицасы;

Кейбір түйіндерге арналған бұтақ сызбасы.

Негізгі ұсынылатын әдебиеттер:

1. Жандаулетова, Ф. Р. Охрана труда : учебник для вузов / Ф.Р. Жандаулетова, Т.Е. Хакимжанов, Т.С. Санатова; МОН РК, НАО АУЭС. - Алматы : АУЭС, 2019. - 399 с.

2. МЕСТ 12.1.005-88 «Жұмыс аймағындағы ауа, жалпы санитарлық-гигиеналық талаптар» – Еңбекті қорғау стандарттарының жүйесі. Жұмыс аймағының ауасына жалпы санитарлық және гигиеналық талаптар. КСРО Денсаулық сақтау министрлігі, Бүкілодықтық кәсіподақтардың орталық кеңесімен 1988 жылы енгізілген.

3. Абикенова А.А., Санатова Т.С. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Пожарная профилактика» в выпускных работах для всех специальностей. Бакалавриат - Алматы: АИЭС, 2009. - 32 с.

4. Т.Е.Хакимжанов, Ф.Р.Жандаулетова. БЖД. Дипломное проектирование. Расчет воздухопровода и выбор вентилятора. Методические указания по выполнению раздела в дипломных проектах для студентов всех форм обучения – Алматы: АИЭС, 2006. - 24 с.

5. Санитарные правила и нормы. СанПиН 2.1.8/2.24.1383-03. Электромагнитные излучения радиочастотного диапазона (ЭМИ РЧ). Госкомсанэпиднадзор России. – М.,2003.

6. Санатова Т.С., Хакимжанов Т.Е.. Охрана труда и основы безопасности жизнедеятельности. Защита от воздействия электромагнитных полей радиочастотного диапазона. Методические указания и задания к расчетно-графической работе для студентов специалистов 5В071900 - Алматы: АИЭС, 2010 - 33 с.

7. Бітіруші жұмысының экономикалық бөлімі үшін әдістемелік нұсқаулар. Радаиотехника, электроника және телекоммуникация мамандығының барлық оқу түрінің студенттеріне арналған. Алматы, 2009.

8. Зыков А. А. Основы теорий графов. М.: Наука, 1987г.

9. Евстигнеев В. А. Применение теорий графов в программировании / под редакцией Ершова А. П. –М: Наука, 1985 г

10. Салғараева Г. И. Графтар теориясы: Алматы: ЖШС «Дәуір», 2013. – 256-бет.

Жоба бойынша жобаның бөлімдеріне қатысты белгіленген кеңесшілер

Бөлімдері	Кеңесшілері	Мерзімі	Қолы
Негізгі бөлім	Хизирова М. А	09.05.2020	
Есептеу техникасын қолдану	Хизирова М. А	09.05.2020	
Өміртіршілік қауіпсіздік бөлімі	Жандаулетова Ф.Р	01.05.2020	
Экономика бөлімі	Тузелбаев Б.И	26.04.2020	
Нормабақылау	Накисбекова Б. Р		

Диплом жобасын дайындау
КЕСТЕСІ

Бөлімдердің атауы, әзірленетін мәселелердің тізімі	Ғылыми жетекшіге ұсыну мерзімдері	Ескерту
Графтар теориясының элементтері	24.02.2020	орындалды
Құрылымның графтар көмегімен бейнеленген негізгі сипаттамалары	03.03.2020	орындалды
Желінің берілген екі түйінімен мүмкін болатын жолдарды құру	4.03.2020	орындалды
Флойд-Воршел алгоритмі	11.03.2020	орындалды
Форд Беллман алгоритмі	19.03.2020	орындалды
Дейкстр алгоритмі	27.03.2020	орындалды
Дейкстрдің сейрек графикалық алгоритмі	04.04.2020	орындалды
Бағдарламма жасау барысында қолданылатын компоненттер және файлдар	11.04.2020	орындалды
Графты сақтау және жүктеу	19.04.2020	орындалды
Дейкстрдің сейрек графикалық алгоритмін қайта есепке ала отырып, анализ жасау	19.04.2020	орындалды
Өміртіршілік қауіпсіздігі бөлімі	26.04.2020	орындалды
Өміртіршілік қауіпсіздігінің техникалық шешімі	27.04.2020	орындалды
Есептеу бөлімі	29.04.2020	орындалды
Техникалық-экономикалық бөлімі	03.05.2020	орындалды
Қаржылық жоспар	03.05.2020	орындалды
Кіріспе және қорытынды бөлімдерін жазу	04.05.2020	орындалды

Тапсырманың берілген уақыты «17» ақпан 2020 ж.

Кафедра меңгерушісі _____ (Темырканова Эльвира Кадылбековна)
(қолы) (Т.А.Ж.)

Жобаның
ғылыми жетекшісі _____ (Хизирова Мухаббат Абдисаттаровна)
(қолы) (Т.А.Ж.)

Орындалатын тапсырманы
қабылдаған студент _____ (Жансай Қыдырәлі Қалманұлы)
(қолы) (Т.А.Ж.)

Андатпа

Бұл дипломдық жобада графтарды зеріттеп, графикалық түйіндер арасындағы ең қысқа жолдарды табуға арналған алгоритмдерді қарастырып, Дейкстра алгоритмін программалық тілде моделдейді.

Жобада желіні суреттеудің, талдаудың және синтездеудің математикалық түсінігін қысқаша бере отырып, графтар теориясының элементтері және де құрылымның графтар көмегімен бейнеленген негізгі сипаттамаларын мысалдармен көрсетеді. Желінің берілген екі түйінімен мүмкін болатын жолдарды құрудың бірнеше әдістерін қолданылды.

Сонымен қатар, өміртіршілік қауіпсіздігімен, жобаның бизнес жоспары құрып, осы жобаға кеткен қаражаттың қайту мерзімі есептелді.

Аннотация

В данной дипломной работе изучается графы, рассматривается алгоритмы поиска кратчайших путей между узлами и моделируется алгоритм Дейкстры на языке программирования.

Проект обобщает математическое понимание описания, анализ и синтез сети, иллюстрируется элементы теории графов и основные характеристики структуры, представленной графами. Несколько методов были использованы для создания возможных путей с этими двумя узлами сети.

Кроме того, был разработан бизнес-план проекта и рассчитан срок окупаемости проекта, а также были приведены расчеты по безопасности жизни деятельности.

Abstract

In this thesis, we study graphs, consider algorithms for finding the shortest paths between nodes, and model the Dijkstra algorithm in a programming language.

The project summarizes the mathematical understanding of the description, analysis and synthesis of the network, illustrates the elements of graph theory and the main characteristics of the structure represented by graphs. Several methods have been used to create possible paths with these two network nodes.

In addition, a project business plan was developed and the payback period of the project was calculated, as well as calculations were made on the safety of life of the activity.

Мазмұны

Кіріспе	8
1 Желіні суреттеудің, талдаудың және синтездеудің математикалық аппараты	10
1.1 Графтар теориясының элементтері.	10
1.2 Құрылымның графтар көмегімен бейнеленген негізгі сипаттамалары....	11
1.3 Желінің берілген екі түйінімен мүмкін болатын жолдарды құру.....	14
2 Қысқа ара қашықтықты табу әдістері	16
2.1 Флойд-Воршел алгоритмі.....	18
2.2 Форд Беллман алгоритмі	18
2.3 Дейкстр алгоритмі.....	18
2.4 Дейкстрдің сейрек графикалық алгоритмі	19
3 Дейкстр алгоритмін Delphi программалық тілінде моделдеу	25
3.1 Бағдарламма жасау барысында қолданылатын компоненттер және файлдар.....	25
3.2 Графты сақтау және жүктеу.....	30
3.3 Дейкстрдің сейрек графикалық алгоритмін қайта есепке ала отырып, анализ жасау	34
4 Өміртіршілік қауіпсіздігі	39
4.1 Еңбек шартын талдау.....	39
4.2 Өміртіршілік қауіпсіздігінің техникалық шешімі	44
4.3 Өміртіршілік қауіпсіздігі бөлімі бойынша қорытынды	50
5 Техникалық-экономикалық бөлім	51
5.1 Жұмыстың мақсаты	51
5.2 Жұмыстың орындалу бағдарламасы	51
5.3 Қаржылық жоспары	51
5.4 Экономика бөлімі бойынша қорытынды	61
Қорытынды	62
Әдебиеттер тізімі	63

Кіріспе

Дипломдық жұмыстың мақсаты: Дейкстр алгоритмін зерттеп, оны программалық тілде моделдеу және жасылынған программалық бағдарламаны компанияларға ұсыну.

Дейкстр алгоритмі (немесе Дейкстр ең қысқа жолының бірінші алгоритмі, SPF алгоритмі) - бұл графикалық түйіндер арасындағы ең қысқа жолдарды табуға арналған алгоритм, мысалы, кез келген телекоммуникациядағы желі. Оны ғалым Дейкстра 1956 жылы ойлап тауып, үш жылдан кейін шығарды. Дейкстр алгоритмі бағанның бір шыңынан басқаларына дейінгі қысқа жолды табады. Алгоритм теріс салмақтың қабырғалары жоқ графтар үшін ғана жұмыс істейді. Негізінде қысқа жолды табатын бірнеше алгоритмдердің түрлері бар, олар: Флойд-Воршел, Форд Беллман және т.б.. Бұл алгоритмді зерттеу себебіміз – Дейкстр алгоритмі телекоммуникациялық салада кең қолданылады.

Графикте берілген бастапқы түйін үшін алгоритм осы түйін мен кез келген басқа арасындағы ең қысқа жолды табады. Ол сонымен қатар мақсатты түйінге ең қысқа жол анықталғаннан кейін алгоритмді тоқтату арқылы бір түйіннен бір тағайындалған түйінге дейінгі қысқа жолды табуға қолданылады. Мысалы, егер графикалық түйіндер қалаларды бейнелесе және жол жиегінің құны тікелей жолмен байланысқан жұп қалалар арасындағы қашықтықтың қозғалысын білдірсе, Дейкстр алгоритмін бір қала мен басқа қалалар арасындағы ең қысқа жолды табуға қолдануға болады. Нәтижесінде, қысқа жолдар алгоритмі желілік бағыттау хаттамаларында, ең алдымен IS-IS (Inter-Gateway System) және ең қысқа жолда (OSPF) кеңінен қолданылады. Ол сонымен қатар Форд Беллман сияқты басқа алгоритмдерде әдеттегідей қолданылады.

Дейкстр алгоритмінде белгілері, оң сандары немесе нақты сандары қолданылады, олар қатаң әлсіз реттелген. Бір қызығы, Дейкстр кез-келген түрде белгіленген белгілерді пайдалануға жалпылануы мүмкін, егер олар қатаң жартылай белгіленген болса және келесі белгілер біркелкі төмендемейтін болса. Бұл жалпылау генеральды Дейкстр қысқа жол алгоритмі деп аталады.

Біз бастайтын түйінге бастапқы түйін деп аталсын. Y түйінінің қашықтығы бастапқы түйіннен Y-ге дейінгі қашықтық болсын. Дейкстр алгоритмі алғашқы қашықтық мәндерін тағайындайды және оларды кезең-кезеңімен жақсартуға тырысады.

Барлық түйіндерді ескерілмеген етіп белгілеңіз. Көрінбейтін барлық түйіндердің жиынтығын жасаңыз, олар қарастырылмаған жиындар деп аталады. Әр түйінге қашықтықтың алдын-ала мәнін беру: оны бастапқы түйін үшін нөлге, ал қалған барлық түйіндерге шексіздікке қойыңыз. Бастау түйінін ағымдағы етіп орнатыңыз.

Ағымдағы түйін үшін оның барлық күтілмеген көршілерін қарастырып, олардың ағымдағы түйін арқылы өтетін қашықтықтарын есептеңіз. Жаңадан есептелген алдын-ала қашықтықты ағымдағы орнатылған нүктемен

салыстырып, кішісін тағайындаңыз. Біз аяқтағаннан кейін, ағымдық түйіннің барлық ескерілмеген көршісін қарастырып, ағымдық түйінді барған деп белгілеп, оны шақырылмаған жиыннан алып тастаңыз. Егер тағайындалған түйін барған болса (екі нақты түйін арасындағы бағдарлауды жоспарлау кезінде), немесе Шақырылмаған жиынтықтағы түйіндердің арасындағы ең аз қашықтық шексіздік болып табылады (толық турды жоспарлау кезінде, ол бастапқы түйін мен байланыссыз түйіндер арасында байланыс болмаған кезде пайда болады), тоқтайды.

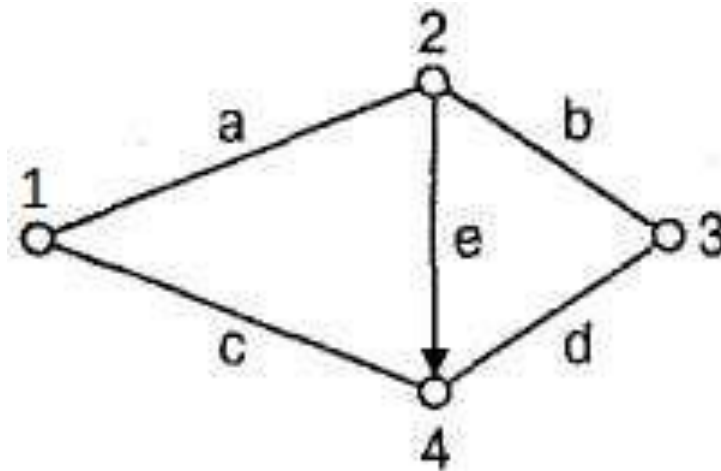
1 Желіні суреттеудің, талдаудың және синтездеудің математикалық аппараты

1.1 Графтар теориясының элементтері

Тораптың топологиялық құрылымының математикалық моделі ретінде граф түріндегі модель жиі қолданылады (сур. 1.1).

Әдетте граф төбелері (1, 2, 3, 4) цифрларымен белгіленеді және КТ және/немесе ШП-мен сәйкестендіріледі, ал граф қабырғалары - (a, b, c, d, e) әріптерімен белгіленеді де, байланыс каналдарына сәйкес келеді.

Символикалық формада графтар $G(A, B)$ түрінде белгіленеді, мұндағы G белгісі осы түсінікке сәйкес логикалық мазмұнды көрсетеді;



1.1 сурет – Желі құрылымының графы

$A = \{a_1, a_2, \dots, a_N\}$ – граф төбелерінің жиыны; $B = \{b_{ij}\}$ - a_i және a_j төбелері арасындағы қабырғалар жиыны. Граф төбелері көршілес деп аталады, егер олар қабырғамен байланыстырылса. Қабырғалар оиентарленген немесе бағытталған (е қабырғасы) және ориентирленбеген немесе бағытталмаған (ребра a, b, c, d) болуы мүмкін. Ориентирленген қабырға бір жақты каналға сәйкес келеді, ал ориентирленбеген қабырға – екіжақты каналға сәйкес болмақ.[12]

Графтың үш түрі бар:

1) ориентирленген графтар, барлық қабырғалары ориентирленген;
2) ориентирленбеген графтар, ориентирленген қабырғаларды қамтымайды;

3) көршілес типті графтар, онда ориентирленген де, ориентирленбеген де қабырғалар бар.

Әр қабырғаға «салмақ» меншіктеледі – сан немесе сандар жиыны, ол берілген қабырғаның қандай да бір болмасын қасиетін сипаттайды. Салмақ ретінде мыналарды алуға болады, мысалы, канал ұзындығы, өткізгіштік мүмкіндігі, ақпаратты беру жылдамдығы, стандартты каналдар саны, сенімділігі, құны және т.б. Граф төбелеріне де белгілі бір салмақ меншіктелуі мүмкін.

Кіріс немесе шығыс (инциденттік) қабырғалар саны $r(a_i)$ түйінінің рангы деп аталады, мұндағы i – түйін нөмірі. $r(a_1) = 2$, $r(a_2) = 3$. 1 ранг түйіні түйықталған болып табылады, себебі ол арқылы ешқандай жол өте алмайды.

a_i түйінінен a_j түйініне m_{ij} жолы – бұл қабырғалардың реттелген жиыны, a_i түйінінде басталып, a_j түйінінде аяқталады. Жол үшін әрбір алдыңғы қабырғаның аяғы келесі қабырғаның басымен сәйкес келеді. Жолдар өздігінен қиылыспайтын болу керек, яғни бір түйін арқылы екі рет өтпеу керек. Граф үшін 1 және 3 төбелер арасында үш жол болады: ab , cd , aed . Осы төбелер арасындағы жолдар жиыны $m_{13} = ab$ и cd және aed . Қабырғалар сияқты жолдар да бағытталған және бағытталмаған болуы мүмкін.

$r(m_{ij})$ жолының рангы дегеніміз осы жолға кіретін қабырғалар саны. Жолдың минимальды рангы 1-ге тең, мысалы $r(m_{12}) = 1$, ал максимальды - N – 1-ге тең, мұндағы N – граф төбелерінің саны, бұл жағдайда жол барлық төбе арқылы өтетін болады.

Бір төбеден басталып сол төбеде аяқталатын жол контур (цикл) деп аталады.

h байланыстылығы дегеніміз төбелердің барлық жұптарының арасындағы тәуелсіз жолдардың минимальды саны. Граф үшін $h = 2$.

1.2 Құрылымның графтар көмегімен бейнеленген негізгі сипаттамалары

Қалааралық және халықаралық байланыстардың қосымша торабының құрылымын құру үшін беріліс жолын анықтау қажет. Екінші торап үшін беріліс жолы – белгілі бір талап етілген сапаға жауап беретін, ақпараттарды (электросигналдар, электртолқындар – эфир, жарық сәулелері, рентген-сәулесі) беруге арналған сәйкес сфера. Беріліс жолы бастапқы желі нүктесінде басталып, аяқталады, онда энергияны енгізуге және шығаруға мүмкіндік алуға болады. Мұндай нүктелерге тораптық түйіндер (сонымен бірге ауыстырып қосу пункті), бастапқы желінің шеткі станциялары, бастапқы жергілікті тораптың шығыстары жатады.

Сызық (желі, линия) – бастапқы торап үшін қолданылатын кейбір жол, өзінің құрамына оған екі жағынан қосылған екінші (қосымша) тораптың техникалық құрылғыларын қамтиды.

Торап құрылымын анықтау мыналарға байланысты:

1) бастапқы қалааралық тораптың шеткі түйіндерінің орналасқан орны берілген;

2) осы түйіндердің саны және осы түйіндердегі құрылғылар типтері анықталған;

3) есептеу бастапқы тораптағы (орны сол сәтте әлі анықтала қоймаған) басқа шеткі түйіндердің пайда болу мүмкіндігін ескере отырып жүргізіледі.

Қалааралық байланыстың бірінші торабы берілген тиянақты түйіндер негізінде, немесе берілген тиянақты территорияда дамиды. Бірінші құрылым сондықтан да бірінші тораптың графтық моделі деп аталады, ал екіншісі қалааралық байланыстың территориялық-бағытталған торабы деп аталады.

Екінші (қосымша) каналдарының шоғырын ұйымдастыру жолын және трассасын анықтау. Желі түйіндері арасындағы мүмкін болатын жолдарды анықтаудың бірнеше тәсілдері бар.

Соның ішіндегі ыңғайлысы және жеңіл программаланатыны – жол бұтағын құру болып табылады. Желі түйіндері өзара жолмен байланысады, оны бұтақ түрінде жеңіл көрсетуге болады.

Бұтақ – циклсыз жекелеген граф.

Маршрут – төбелер мен қабырғалардың ақырғы кезектесетін тізбегі.

Маршрут ашық және тұйықталған болуы мүмкін.

Желіні құрған кезде маршрут үнемі ашық болады.

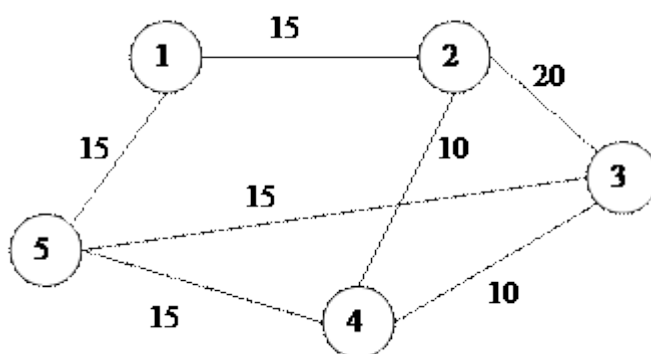
Жолдағы қабырғалар саны жол ұзындығы деп аталады. Жол ұзындығы осы жолға кіретін ұзындықтар қосындысымен анықталады.

Жол сыйымдылығы – осы жолға кіретін тармақтардың минимальды сыйымдылығы.

Бұтақты құру алгоритмі. Бұтақты беру үшін байланыс матрицасын құру қажет, бұл – екі төбенің байланысын анықтайтын матрица, себебі желі – тармақсыз ақырғы граф.

Берілген желі үшін байланыс матрицасын құралық.

$$i = 5, j = 7$$



1.2 сурет – Граф арқылы көрсертілген желі

B – байланыс матрицасы, желіде қанша қатар және қанша баған бар. Қосымша диагональ – түйіндердегі (қажет емес) тармақтар.

$$B = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} - & 1 & 0 & 0 & 1 \\ 1 & - & 1 & 1 & 0 \\ 0 & 1 & - & 1 & 1 \\ 0 & 1 & 1 & - & 1 \\ 1 & 0 & 1 & 1 & - \end{bmatrix} \end{matrix}$$

1. $k=1, 2, 3, 4, 5$ түйінін таңдаймыз (ол қажетті бұрышпен тікелей байланыспаған) k - шы қатарды таңдаймыз;

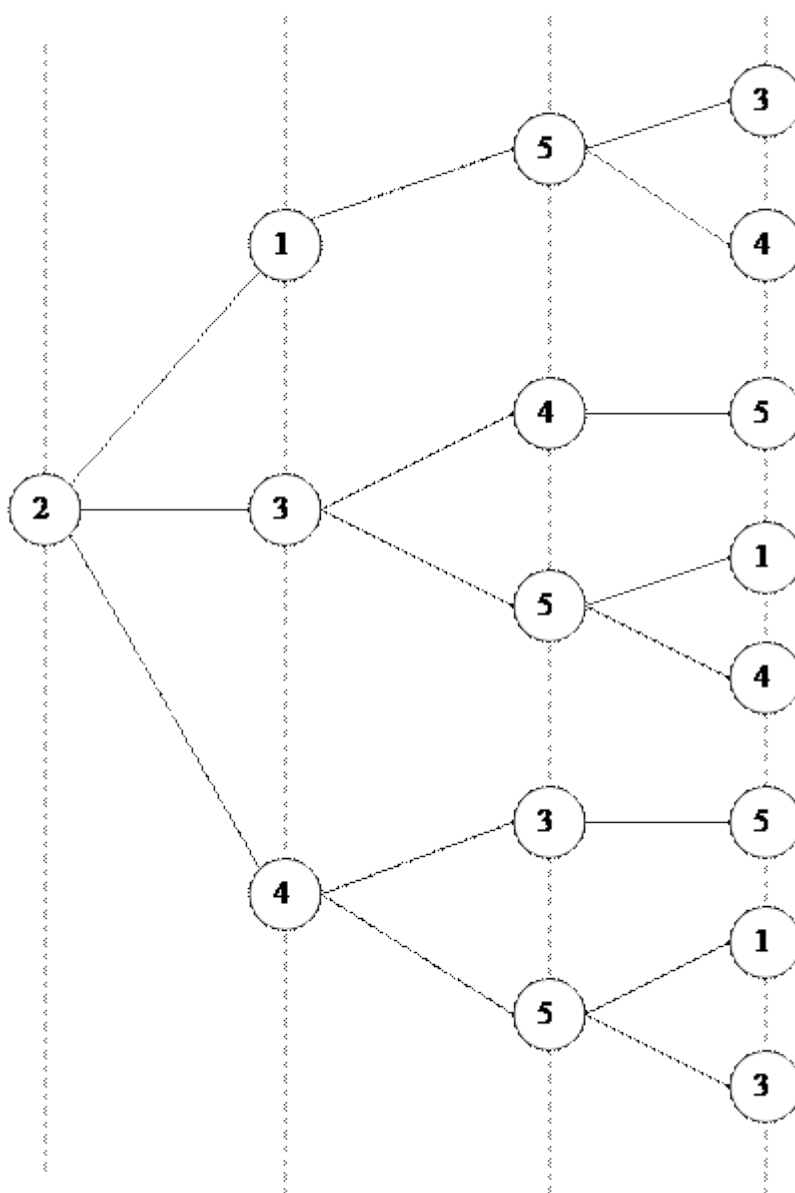
2. осы жолдың бағандар нөмірін теріп жазамыз, ондағы матрица элементтерінің мәндері $b_{ij}=1$ және k түйінімен түзілген бірінші рангты түйіндердің ішкі жиынын аламыз;

3. осындай ішкі жиынды екінші рангты түйіндерден де аламыз;

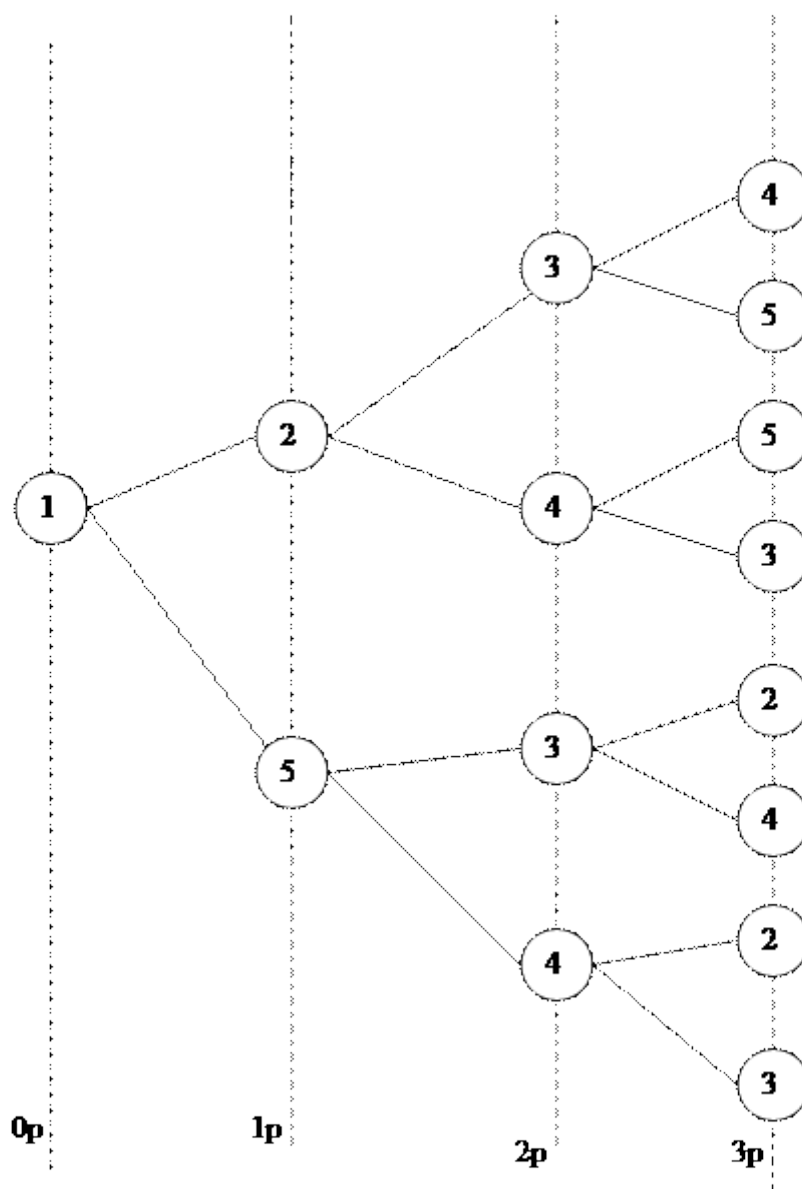
4. бұтақты құрған кезде бағандар нөмірлері (осыған сәйкес ішкі жиын түзілген болады);

5. құру $n=3$ (байланыс торабы үшін) рангына дейін жалғасады, мұнда үшінші рангтан жоғарылары цикл түзе бастайды;

Жол бұтағын иемдене отырып, тиянақталған түйіндер арасындағы мүмкін болатын барлық жолдарды жеңіл табуға болады.



1.3 сурет – 2-ші түйінге арналған бұтақ



1.4 сурет – 1-ші түйінге арналған бұтақ

1.3 Желінің берілген екі түйінімен мүмкін болатын жолдарды құру

Жолдар бұтағы мүмкін болатын жолдардың жиыны береді немесе тораптың берілген түйіндері арасындағы байланыстарды береді, сондықтан да бұтақты иемдене отырып, келесі жолдар жиынтығын құруға болады.

$$M_{25} = \{2.1.5; 2.3.5; 2.4.5; 2.3.4.5; 2.4.3.5\}$$

$$M_{14} = \{1.2.4; 1.5.3; 1.2.4.3; 1.5.4.3\}$$

$$M_{13} = \{1.2.3; 1.5.3; 1.2.4.3; 1.5.4.3\}.$$

Торап түйіндері арасындағы жолдар тізімін жолдар кестесі түрінде жазуға болады. А мәні k – қатардан және m – бағаннан тұрады. А матрицасы осы алгоритмді программалық жүзеге асыруда қолданылады.

$A(k*m)$, k – жолдар саны; m – осы жолдағы тармақтар саны.

Матрица элементі $a_{ij}=1$, егер β_i тармағы μ_i жолына жатса. Егер тармақ ол жолда жатпаса, онда a_{ij} – элементі жазуда қарапайымдылық үшін кестеге енгізілмейді.

1.1 кесте – Екінші шыңнан бесінші шыңға дейінгі тармақтар мен жолдар жазбалары

$\beta_j \backslash \mu_i$	β_{12}	β_{15}	β_{23}	β_{24}	β_{35}	β_{45}	β_{25}	β_{34}
$\mu_1(2,5)$								
$\mu_2(2,5)$								
$\mu_3(2,5)$								
$\mu_4(2,5)$								
$\mu_5(2,5)$								

$\mu_1(2,5)=2-1-5$ - маршруттар

$\mu_2(2,5)=2-3-5$

$\mu_3(2,5)=2-4-5$

$\mu_4(2,5)=2-3-4-5$

$\mu_5(2,5)=2-4-3-5$

1.2 кесте – Бірінші шыңнан төртінші шыңға дейінгі тармақтар мен жолдар жазбалары

$\beta_j \backslash \mu_i$	β_{12}	β_{15}	β_{23}	β_{24}	β_{35}	β_{45}	β_{34}
$\mu_1(1,4)$							
$\mu_2(1,4)$							
$\mu_3(1,4)$							
$\mu_4(1,4)$							

$\mu_{14}=\{1, 2, 4; 1, 5, 4; 1, 2, 3, 4; 1, 5, 3, 4\}$

1.3 кесте – Бірінші шыңнан үшінші шыңға дейінгі тармақтар мен жолдар жазбалары

$\beta_j \backslash \mu_i$	β_{12}	β_{15}	β_{23}	β_{24}	β_{35}	β_{45}	β_{34}
$\mu_1(1,3)$							
$\mu_2(1,3)$							
$\mu_3(1,3)$							
$\mu_4(1,3)$							

$\mu_{13}=\{1, 2, 3; 1, 5, 3; 1, 2, 4, 3; 1, 5, 4, 3\}$

Байланыс каналы – байланыстың тиянақталған түйіндері арасында сигналдардың өтуін қамтамасыз ететін техникалық жабдықтардың, программалық интерфейстердің және хаттамалардың жиынтығы, каналдардың мынадай параметрлері бар: өткізу мүмкіндігі, көлемі, уақыты, жиілігі, деңгейі.

Каналдар шоғыры – түрлі параметрлерде беруді қамтамасыз ететін екі немесе одан көп каналдардың жиынтығы.

Сыйымдылық – бұл каналдар саны, ақпарат саны, ағындар, тапсырыстар және абоненттер талаптарының саны.

Тораптың екі түйінінің арасындағы каналдар үймелерінің оптимальды сыйымдылығын есептеу және таңдау тапсырмасы сызықтық есеп түрінде қалыптасқан және сызықтық программалау тәсілімен шешіледі.

$M = \{\mu_i\}$ – жолдар жиынтығы болсын. f_i – μ_i жолының сыйымдылығы. Салмақ – жолды қолданудың салыстырмалы бүтіндігіне анықтама, ол жол ұзындығына кері пропорционалды таңдалуы мүмкін немесе үлестіруді оптимизациялау критеріі бойынша таңдалуы мүмкін.

Салмақ ұзындығы бойынша былай таңдалады

$$C_i = k \left(\frac{l_{max}}{l_i} \right), \quad (1.1)$$

мұндағы, k – осы тармақтың коэффициенті;

l_{max} – ең ұзын жол – осы жолдағы қабырғалар саны;

l_i – қарастырылатын жол.

t2 Қысқа ара қашықтықты табу әдістері

Қысқа ара қашықтықты табу - графикте екі нүктенің (шыңдардың) арасындағы ең қысқа жолды (тізбекті) көрсерту, онда тізбекті құрайтын қабырғалардың салмағының қосындысы азаяды. Қысқа ара қашықтықты табу әдістері - бұл графикалық теориядағы маңызды классикалық мәселелердің бірі. Бүгінгі күні оны шешудің көптеген алгоритмдері бар. Бұл тапсырманың басқа атаулары бар: жолдың минималды мәселесі немесе ескірген нұсқада - тоқу мәселесі. Бұл тапсырманың маңыздылығы оның әртүрлі практикалық қолдануымен анықталады. Мысалы, GPS навигаторлары екі қиылыстың арасындағы ең қысқа жолды іздейді. Жол қиылыстары шыңдар ретінде жұмыс істейді, ал жолдар олардың арасындағы қабырғалар болып табылады. Егер қиылыстар арасындағы жолдардың ұзындығының қосындысы аз болса, онда табылған жол ең қысқа болады. Графикте ең қысқа жолды табу міндеті басқарылмаған, бағдарланған немесе аралас график үшін анықталуы мүмкін. Әрі қарай, мәселенің шешілуін қарапайым емес түрде басқарылмайтын график үшін қарастырамыз. Аралас және бағытталған график үшін жиектердің бағытын да ескеру қажет.

Қысқа ара қашықтықты табу әдістерінің әртүрлі тұжырымдары бар:

Берілген межеге жетудің ең қысқа жолы. Графиктің шыңдарының әрқайсысында басталатын t нүктесінің ең қысқа жолын табу қажет (t -дан басқа). Графикке жататын әр жиектің бағытын өзгерте отырып, бұл мәселені бір көздің шыңына дейін азайтуға болады. Берілген шыңдар арасындағы қысқа жол мәселесі. Берілген u шыңынан бастап v шыңына дейінгі қысқа жолды табу қажет. Барлық жұп шыңдардың арасындағы ең қысқа жол мәселесі. Әрбір u шыңынан бастап v шыңына дейін ең қысқа жолды табу қажет. Бұл мәселені сонымен қатар бір көздің шыңы есебін шешуге арналған алгоритм көмегімен шешуге болады, бірақ әдетте ол тезірек шешіледі.

Мәселенің әртүрлі нұсқаларында қабырғаның ұзындығының рөлін тек ұзындықтар ғана емес, сонымен бірге уақыт, шығындар, шығындар, жұмсалған ресурстардың мөлшері (материалдық, қаржылық, отын-энергетикалық және т.б.) немесе әр қабырғаның өтуімен байланысты басқа да сипаттамалары бойынша ойнауға болады. Осылайша, проблема көптеген салаларда (информатика, экономика, география және т.б.) практикалық қолдануды табады. Бұл мәселенің әртүрлі тұжырымдары болғандықтан, графикте ең қысқа жолды табу мәселесін шешудің ең танымал алгоритмдері бар: Дейкстра алгоритмі графиктің шыңдарының бірінен басқаларына дейінгі ең қысқа жолды табады. Алгоритм тек теріс салмағы жоқ сызбалар үшін жұмыс істейді. Беллман-Форд алгоритмі салмақталған графиктен графиктің бір шыңынан басқаларына дейінгі қысқа жолдарды табады. Шеттердің салмағы теріс болуы мүмкін. А * іздеу алгоритмі диаграммадағы ең жақсы сәйкестікті іздеу алгоритмін қолдана отырып, бір шыңнан (бастапқы) екіншісіне (мақсатты, түпкілікті) ең аз шығындар бағытын табады. Флойд-Воршелль алгоритмі өлшенген бағытталған графиктің барлық шыңдары арасындағы ең қысқа жолдарды табады. Джонсонның алгоритмі өлшенген бағытталған графиктің барлық жұп шыңдары арасындағы ең қысқа жолдарды табады.

Ли алгоритмі (толқындық алгоритм) кеңейтілген бірінші іздеу әдісіне негізделген. Аралық шеттердің (шеттердің) минималды санын қамтитын графиктің s және t шеттерінің арасындағы сызықты табыңыз (t сәйкес келмейді). Негізгі қолдану чиптер кристалдарында және баспа схемаларында электрлік қосылыстарды бақылау болып табылады. Сондай-ақ, стратегиялық ойындарда картадағы ең қысқа қашықтықты табу үшін қолданылады. Кильдалль алгоритмі негізінде ең қысқа жолды іздеңіз. Жұмыс (Черкацкий және басқалар, 1993) осы мәселені шешудің тағы бірнеше алгоритмдерін ұсынды. Көптеген ойын жасаушылар (немесе бағдарламалауды жақсы көретін адамдар) ең қысқа жолдардағы мәселелерді шешетін осы төрт маңызды алгоритмді тыңдағысы келетіні сөзсіз.

График бірнеше нүктелер (шыңдар) деп аталады, олардың бірнеше жұбы сегменттермен (жиектермен) жалғанады. Егер осы сегменттер бойынша кез-келген басқа шыңнан мүмкін болса, график қосылады. Цикл дегеніміз - сол шыңнан басталатын және аяқталатын сызбаның жиегіндегі жол. Егер әрбір жиегі белгілі бір санға (салмаққа) сәйкес келсе, тағы бір график салмақты деп аталады. Бір шыңдарды қосатын екі жиек болмауы керек.

Алгоритмдердің әрқайсысы салмақты жалғанған қысқа жолдардың кейбір мәселелерін шешеді. Бір шыңнан екіншісіне дейінгі қысқа жол - бұл біз өткен шеттердің салмағының қосындысы аз болатындай шеттердегі жол.

Түсінікті болу үшін мен нақты өмірде осындай тапсырманы мысал келтіремін. Елімізде осы қалаларды жалғайтын бірнеше қала мен жолдар болсын. Сонымен қатар, әр жолдың ұзындығы бар. Сіз мүмкіндігінше аз саяхаттап, бір қаладан екінші қалаға барғыңыз келеді. Графикте n шыңдары мен m шеттері бар деп болжаймыз. Қарапайымнан күрделіге көшейік.

2.1 Флойд-Воршел алгоритмі

N^3 ретті операциялардың саны үшін әр шыңнан әрқайсысына дейінгі қашықтықты табады. Салмақ теріс болуы мүмкін, бірақ бізде шеттердің салмағының теріс қосындысы бар цикл болуы мүмкін емес (әйтпесе біз оны қалағанымызбен жүре аламыз және әр уақытта мөлшерін азайта аламыз, қызық емес).

$D[0 \dots n-1][0 \dots n-1]$ массивінде i -ші итерация кезінде біз түпнұсқа мәселенің жауабын шектеулермен сақтаймыз, себебі біз жолдардағы «мен өту нүктелері» ретінде мен санынан аз болатын шыңдарды қолданамыз. - 1 (шыңдарды нөлден бастап санаймыз). Итерацияны жіберейік, ал массивті $i+1$ -ге жаңартқымыз келеді. Ол үшін біз әрбір шыңның i -ші шыңын өзара ауыстырып қосуға тырысамыз, егер бұл жауап жақсарса, оны тастап кетейік. Жалпы, біз $n+1$ итерациясын жасаймыз, оны аяқтағаннан кейін кез-келген «алмасу» ретінде қолдануға болады, ал d жиымы жауап болады [11].

2.2 Форд Беллман алгоритмі

Бір шыңнан қашықтығын табыңыз (оған 0 санын берейік) $n * m$ ретіндегі операциялардың саны үшін. Алдыңғы алгоритмдегідей, салмақ теріс болуы мүмкін, бірақ жиектерінің теріс қосындысы бар циклдар болмайды.

Біз $d[0 \dots n-1]$ массивін бастаймыз, онда i -ші итерация кезінде бастапқы мәселеге жауапты i шеттерінен аз болатын жолмен кіруге болатын шектеумен сақтаймыз. Егер j шыңына жетудің мұндай жолдары болмаса, онда $d[j] = 2,000,000,000$ (бұл мүмкін емес тұрақты, “шексіздік” болуы керек). D -нің басында 2000000000 толтырылады. Итерацияны итерация кезінде массивті жаңарту үшін сіз әр шетінен өтіп, оны жалғаған шыңдарға дейінгі қашықтықты жақсартуға тырысуыңыз керек. Қысқа жолдарда цикл болмайды, өйткені барлық циклдар негативті емес, және біз циклды жолдан алып тастай аламыз, және жолдың ұзындығы нашарламайды (сонымен қатар, сіз графикте теріс циклдерді осылай табуға болатындығын атап өткім келеді: біз тағы бір итерация жасаймыз және көрейік пе? кейбір шыңға дейінгі арақашықтық жақсарды ма). Сондықтан ең қысқа жолдың ұзындығы $n-1$ -ден аспайды, демек d -ші итерациядан кейін мәселенің жауабы болады [12].

2.3 Дейкстра алгоритмі

Бір шыңнан қашықтықты табады (0 санын берейік) n^2 ретгі операциялардың саны үшін. Барлық салмақ теріс емес.

Әр итерация кезінде кейбір шыңдар белгіленеді, ал кейбіреуі жоқ. Біз екі массивті бастаймыз: $[0 \dots n - 1]$ белгісі - рас, егер шыңы белгіленген болса, Жалған болса, $d[0 \dots n - 1]$ - әрбір шың үшін ең қысқа жолдың ұзындығы тек белгіленген шыңдар бойымен өтеді, өйткені «өткізу нүктелері» сақталады. . Белгіленген шыңдар үшін d -де көрсетілген ұзындыққа жауап болатын инвариант қолданылады. Біріншіден, тек 0 шыңы белгіленеді, ал $g[i]$ x -ке тең болады, егер 0 мен i салмақтың x жиегіне жалғанған болса, егер олар шеттермен қосылмаған болса, $2\,000\,000\,000$ -ға тең, ал 0-ге $i = 0$.

Әр итерация кезінде белгіленбегендер қатарынан d -нің ең кіші мәні бар шыңын табамыз, ол v шыңы болсын. Онда $d[v]$ мәні v -ге жауап береді. Біз дәлелдейміз. Мысалы, 0-ден v -ге дейінгі ең қысқа жол белгіленген нүктелермен ғана емес, сонымен бірге $d[v]$ -ден қысқа болады. Осы жолдағы бірінші белгіленбеген шыңға жетіңіз, оны шақырыңыз. Жолдың жүріп өткен бөлігінің ұзындығы (0-ден u -ге дейін) $d[u]$. $len \geq d[u]$, мұндағы len - ең қысқа жолдың ұзындығы 0-ден v -ге дейін (өйткені теріс жиектер жоқ), бірақ біздің болжамымыз бойынша $d[v]$ -ден аз. Сондықтан $d[v] > len \geq d[u]$. Бірақ содан кейін v оның сипаттамасына сәйкес келмейді - ол белгіленбегендер арасында $d[v]$ ең кіші мәні болмайды. Қарама-қайшылық.

Енді v шыңын батыл түрде белгілеп, d қайта есептеңіз. Біз мұны барлық шыңдар белгіленіп, d мәселеге жауап болғанша жасаймыз.

n итерация бойынша n итерация (v шыңын іздеу үшін), n^2 амалдарының ретін құрайды.

2.4 Дейкстрың сейрек графикалық алгоритмі

Дейкстр алгоритмімен бірдей әрекетті жасайды, бірақ $m \cdot \log(n)$ ретіндегі амалдардың саны үшін. Айта кету керек, $m \cdot n^2$ тәрізді болуы мүмкін, яғни Дейкстр алгоритмінің бұл өзгеруі әрдайым классикалыққа қарағанда тез емес, тек кішкентай m үшін.

Дейкстр алгоритмінде бізге не қажет? Біз d мәні бойынша ең төменгі шекті таба білуіміз керек және кейбір шыңдағы d мәнін жаңарта алуымыз керек. Классикалық іске асыруда біз қарапайым массивті қолданамыз, n амалдарының реті бойынша d -дің ең төменгі шегін табамыз, және 1 амалда жаңартамыз. Біз екілік үйінді қолданамыз (көптеген нысандарға бағытталған тілдерде ол кірістірілген). Үйінді операцияларды қолдайды: үйіндіге элемент қосыңыз (тіркеу (n) операцияларының реті үшін), ең аз элементті табыңыз (1 операция үшін), ең аз элементті жойыңыз ($\log(n)$ операциялары реті үшін), мұндағы n - үйіндідегі элементтер саны.

$D[0 \dots n - 1]$ массивін құрыңыз (оның мәні бұрынғыдай) және q жиынтығы. Үйіндіде біз v және $d[v]$ шыңдарынан жұптарды сақтаймыз (жұптар $d[v]$ -де салыстырылуы керек). Сондай-ақ үйінділерде мылжын элементтері болуы мүмкін. Бұл $d[v]$ мәні жаңартылғандықтан болады, бірақ біз оны өзгерте алмаймыз. Сондықтан, үйінділерде бірдей шыңдары бар, бірақ әр

түрлі d мәндері бар бірнеше элементтер болуы мүмкін (бірақ үйіндіде m шыңдары болмайды, мен бұған кепілдік беремін). Үйіндідегі минималды мәнді қабылдағанда, біз бұл элементтің мылжың екенін тексеріп алуымыз керек. Ол үшін үйіндідегі d мәні мен оның нақты мәнін салыстыру жеткілікті. График жазу үшін екілік массивтің орнына тізімдер массивін қолданамыз.

Белгілері:

- V — бірнеше графтың шыңдары;
- E — бірнеше графтың қабырғалары;
- $w[ij]$ — ij қабырғасының салмағы(ұзындығы);
- a — шың, ізделетін қашықтықтар;
- U — бірнеше a – дан басқа шыңдар;
- $d[u]$ — алгоритм біткеннен кейінгі a -дан u -ге дейінгі ең қысқа жолдың ұзындығына тең;

ұзындығына тең;

- $p[u]$ — алгоритм біткеннен кейінгі a -дан u -ге дейінгі ең қысқа жолды қамтиды.

$d[a] \leftarrow 0, p[a] \leftarrow a$ болсын делік

Барлығы үшін $u \in V$, тек a -дан басқаларына

$d[u] \leftarrow \infty$ жасаймыз

Әзірге $\exists v \notin U$

$v \in U$ — $d[v]$ –ның минималды шыңы

Барлығы үшін $u \notin U, vu \in E$

егер $d[u] > d[v] + w[vu]$ онда

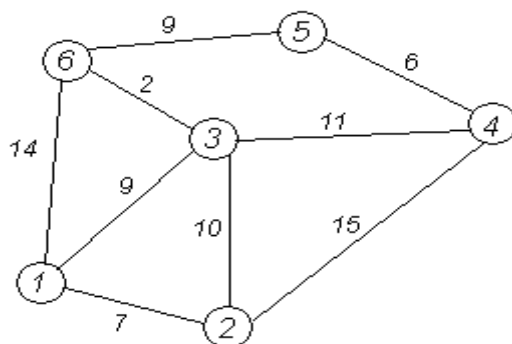
өзгертеміз $d[u] \leftarrow d[v] + w[vu]$

өзгертеміз $p[u] \leftarrow p[v], u$

Қарапайым іске асыруда сандар жиынын $d[i]$ сандарын сақтау үшін қолдануға болады, ал U логикалық айнымалылар жиынын U жиынына жататын элементті сақтау үшін қолдануға болады[10].

Суретте көрсетілген график мысалында алгоритмнің орындалуын қарастырыңыз. (сур. 2.1).

Бірінші шыңнан басқаларына дейінгі ең қысқа қашықтықты табу талап етілсін.

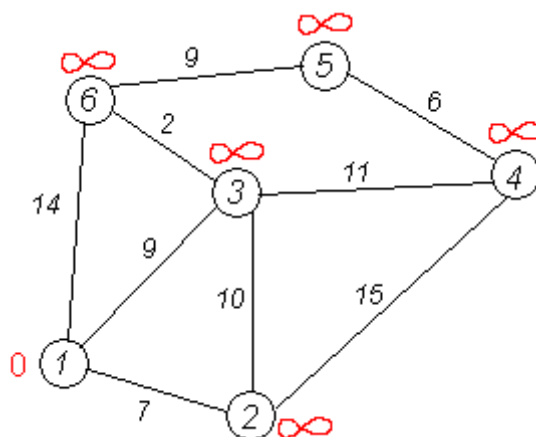


2.1 сурет – Граф арқылы көрсертілген желі

Шеңберлер шыңдарды, жолдар олардың арасындағы жолдарды көрсетеді (сызық жиектері).

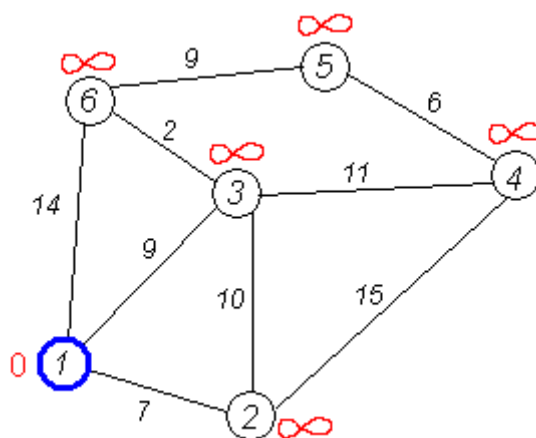
Шеңберлер шыңдардың санын, ал олардың қабырғаларының үстінде олардың салмағы - жолдың ұзындығын көрсетеді.

Жапсырма әр шыңның жанында қызыл түспен белгіленеді - бұл шыңнан 1 шыңға дейінгі қысқа жолдың ұзындығы.



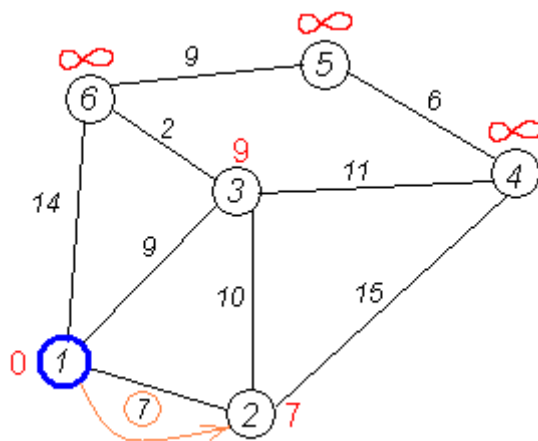
2.2 сурет – Граф арқылы көрсертілген желі

Алғашқы қадам. Ең төменгі жапсырмада 1 шыңы бар. Оның көршілері - 2, 3 және 6 шыңдары.



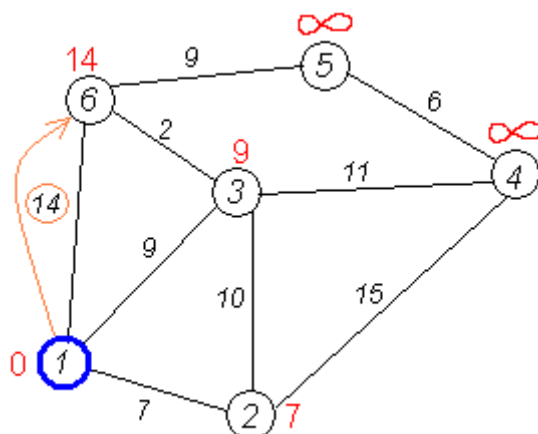
2.3 сурет – Граф арқылы көрсертілген желі

Бірінші шыңның бірінші көршісі - 2-ші шың, өйткені оған баратын жол аз. Оған 1 шыңы арқылы өтетін жолдың ұзындығы 1 шыңы мен 1-ден 2-ге дейінгі жиектің ұзындығына, яғни $0 + 7 = 7$ тең болады. Бұл шыңы 2 шыңының қазіргі белгісінен аз, сондықтан 2-ші шыңның жаңа белгісі 7-ге тең.



2.4 сурет – Граф арқылы көрсертілген желі

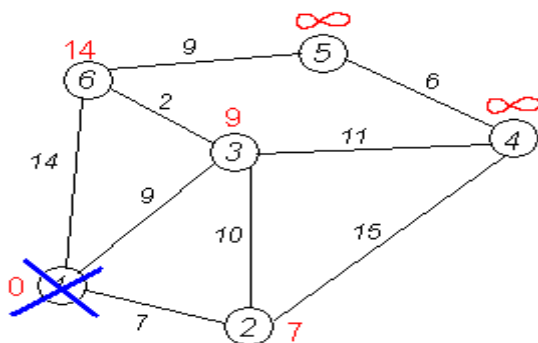
Ұқсас операцияны біз 1-ші шыңның екі көршісімен - 3-ші және 6-шы орынмен жасаймыз.



2.5 сурет – Граф арқылы көрсертілген желі

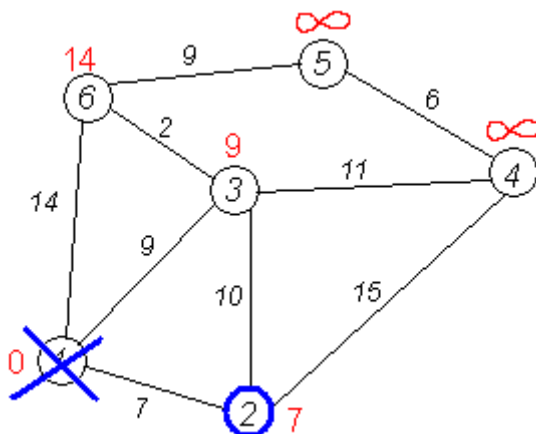
1-шыңның барлық көршілері тексерілген. 1 шыңына дейінгі ең аз қашықтық соңғы болып саналады және қайта қаралмайды.

Графиктен осы шыңға жеткенін атап өту үшін сызыңыз.



2.6сурет – Граф арқылы көрсертілген желі

Екінші қадам. Тағы да біз күтілмеген шыңдардың ең «жақынын» табамыз. Бұл 2 шыңы, 7 белгісі бар



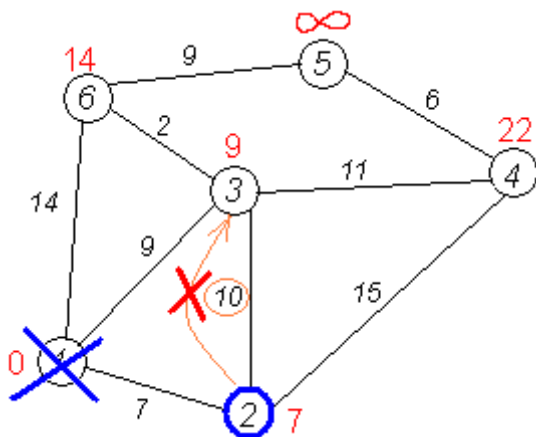
2.7сурет – Граф арқылы көрсертілген желі

Тағы да, біз 2-шыңнан өтуге тырысып, таңдалған шыңның көршілерінің белгілерін азайтуға тырысамыз. 2 шыңдарының көршілері - 1, 3 және 4 шыңдары.

2-шыңның бірінші (көршілес) көршісі - 1-ші шың. 1-ші шың.

Келесі көрші - бұл 3-ші шыңы, өйткені ол ең аз белгіге ие.

Егер сіз оған 2 арқылы кірсеңіз, онда мұндай жолдың ұзындығы 17-ге тең болады ($7 + 10 = 17$). Бірақ үшінші шыңның қазіргі таңбасы - 9, ол 17-ден аз, сондықтан жапсырма өзгермейді.

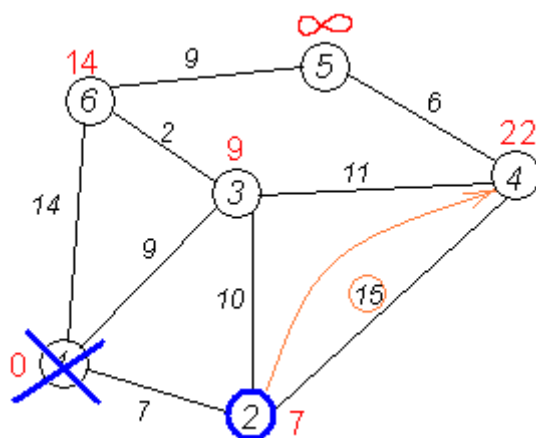


2.8сурет – Граф арқылы көрсертілген желі

Екінші шыңның тағы бір көршісі - 4 шыңы.

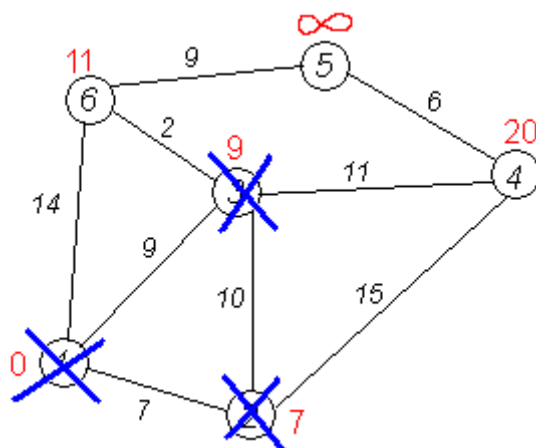
Егер сіз оған 2-ші арқылы кірсеңіз, онда мұндай жолдың ұзындығы 2-ші шыңға дейінгі ең қысқа қашықтықтың және 2 мен 4 шыңдарының арасындағы қашықтықтың қосындысына, яғни 22-ге тең болады ($7 + 15 = 22$).

$22 < \infty$ болғандықтан, 4-тен 22-ге дейін шыңның белгісін қойыңыз.



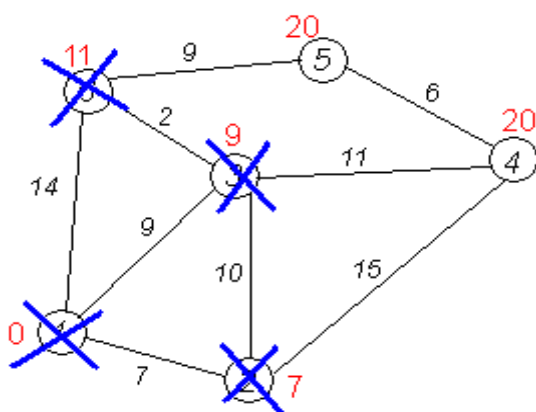
2.9сурет – Граф арқылы көрсертілген желі

Үшінші қадам. Алгоритмнің қадамын 3 шыңын таңдау арқылы қайталаймыз. Оны өңдегеннен кейін келесі нәтижелерге қол жеткіземіз:

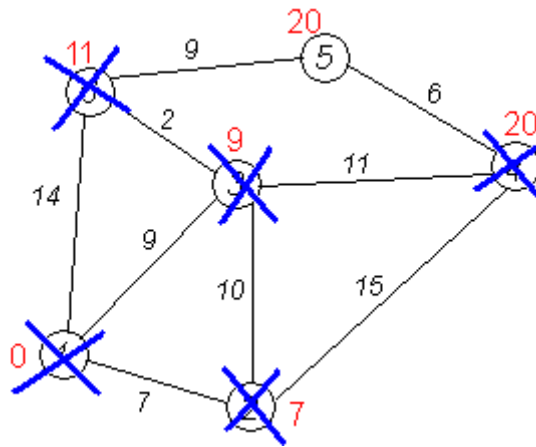


2.10 сурет – Граф арқылы көрсертілген желі

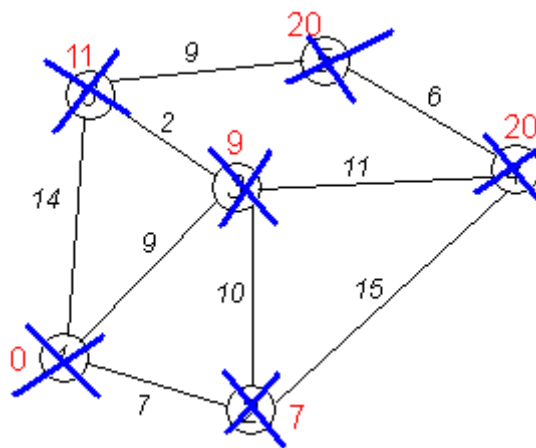
Келесі қадамдар. Қалған шыңдар үшін алгоритм қадамын қайталаймыз. Бұл сәйкесінше 6, 4 және 5 шыңдары болады.



2.11 сурет – Граф арқылы көрсертілген желі



2.12 сурет – Граф арқылы көрсертілген желі



2.13 сурет – Граф арқылы көрсертілген желі

Алгоритмді аяқтау

Алгоритм барлық шыңдарға барған кезде аяқталады.

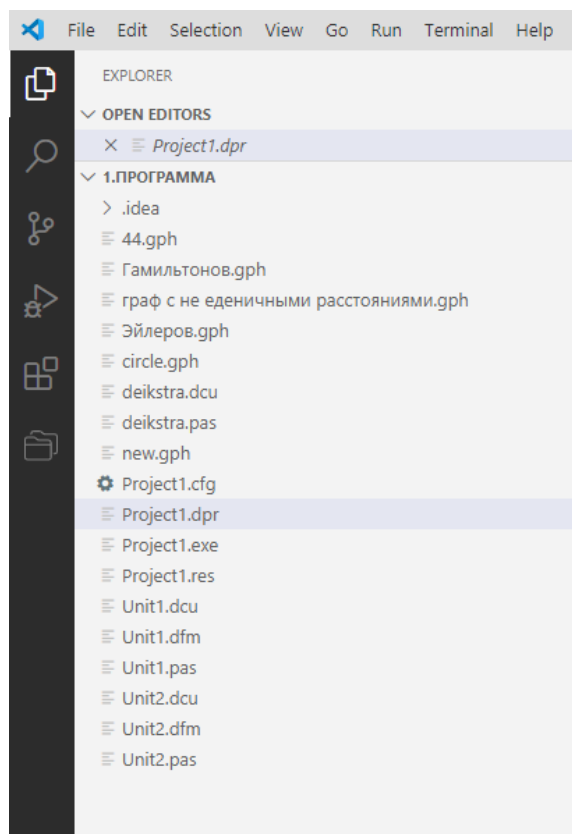
Алгоритмнің нәтижесі соңғы суретте көрінеді: жоғарғы 1-ден 2-ге дейінгі қысқа жол - 7, 3 - 9, 4 - 20, 5 - 20, 6 - 11.

Егер қандай да бір сәтте барлық қарастырылмаған шыңдар шексіздікпен белгіленсе, демек бұл шыңдарға қол жетімсіз болады (яғни, граф ажыратылған). Сонда алгоритмді мерзімінен бұрын аяқтауға болады.

3 Дейкстра алгоритмін Delphi программалық тілінде моделдеу

3.1 Бағдарламма жасау барысында қолданылатын компоненттер және файлдар

Мақсатымыз – бағдарламаны қолдану әдістерін үйретіп, жалпы алгоритм қандай функцияларда қалай жұмыс жасайтынын прогаграммалық блог-сызбалар көмегімен түсіндіру. Және де бағдарламаны жасау барысында қолданылған компоненттерге қысқаша сипаттама беру. Жоғарыда көрсетілген мысалдарды осы бағдарламамен тексеру, және салыстыру.



3.1 сурет – Бағдарламаның негізгі файлдар тізімі

Бағдарламаны барынша қарапайым етіп жасадым, мұнда небәрі 2 Form қолданылады. Оны төменде көрсетілген суретте Project1.dpr файлындағы коды арқылы көруге болады.

```

Project1.dpr - 1.Программа - Visual Studio Code [Administrator]

Project1.dpr
1  program Project1;
2
3  uses
4      Forms,
5      Unit1 in 'Unit1.pas' {Form1},
6      deikstra in 'deikstra.pas',
7      Unit2 in 'Unit2.pas' {Form2};
8
9  {$R *.res}
10
11 begin
12     Application.Initialize;
13     Application.CreateForm(TForm1, Form1);
14     Application.CreateForm(TForm2, Form2);
15     Application.Run;
16 end.
17

```

3.2 сурет – Бағдарламаның Project1.dpr файлындағы код

Алгоритмнің басында бастапқы шыңның қашықтығы нөлге тең болады, ал қалған барлық қашықтық үлкен оң санмен толтырылады (сызбадағы мүмкін болатын жолдан үлкен). Жалаулар массиві нөлдермен толтырылған. Содан кейін негізгі цикл басталады.



3.3 сурет – Блог-сызба

Циклдың әр қадамында біз ең аз қашықтықты және нөлдік жалауша бар шыңды іздейміз. Содан кейін біз жалаушаны 1-ге қойып, оған жақын орналасқан барлық шеттерін тексереміз. Егер ондағы қашықтық ағымдағы шыңға дейінгі қашықтық пен жиіктің ұзындығынан үлкен болса, оны азайтыңыз. Цикл барлық шыңдардың жалаушалары 1-ге тең болғанда немесе барлық шыңдармен 0 жалаушасы болғанда аяқталады. Соңғы жағдай G графы ажыратылған жағдайда мүмкін болады.

Дейкстр алгоритмінің күрделілігі v шыңын табу әдісіне, сонымен қатар көптеген ескерілмеген шыңдарды сақтау әдісіне және жапсырмаларды жаңарту әдісіне байланысты. N шыңдарының санын, ал m - жиіліктің G санын белгілеңіз.

Қарапайым жағдайда, шыңдардың барлық жиынтығы ең төменгі $d[v]$ шыңымен іздеу үшін қарастырылған кезде, ал массив d мәндерін сақтау үшін қарастырылған кезде, алгоритмнің жұмыс уақыты $O(n^2 + m)$ болады. Негізгі цикл n рет реті бойынша орындалады, олардың әрқайсысында минималды табу

үшін n амалдар қажет, сонымен қатар бастапқы графикадағы жиектер санынан аспайтын релаксациялар саны (белгі өзгерістері).

Сызбалар үшін (яғни, m -ден n -ден аз), көрінбейтін шыңдарды екілік үйіндіде сақтауға болады, ал $d[i]$ мәндерін кілт ретінде пайдалануға болады, содан кейін модификацияланған уақытқа қарамастан, одан шынды алу уақыты журналға айналады. Цикл n рет ретпен орындалғандықтан және релаксациялар саны m -дан аспайтындықтан, мұндай жылдамдық $O(n \log n + m \log n)$ құрайды.

Интерфейс - жүйе элементтерінің өзара әрекеттесу құралдары мен әдістерінің жиынтығы.

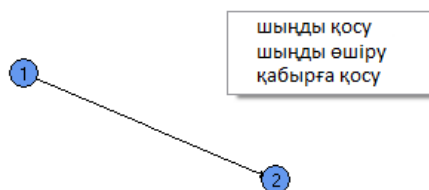
Delphi көптеген мүмкіндіктерге ие. Қоршаған орта Windows белгілері, белгішелер, тіпті диалогтық терезелер сияқты компоненттерді бағдарламалау қажеттілігін болдырмайды. Диалогтық панельдер (мысалы, Файлды таңдаңыз және Файлды сақтаңыз) Delphi-ге тікелей салынған қайта пайдалануға болатын компоненттердің мысалдары болып табылады, бұл осы компоненттерді қолданыстағы тапсырмаға сәйкес жұмыс істеуі үшін осы компоненттерді бейімдеуге мүмкіндік береді. Сондай-ақ, алдын-ала анықталған визуалды және визуалды емес нысандар бар, олардың ішінде батырмалар, деректер нысандары, мәзірлер және диалог терезелері бар. Осы объектілерді қолдана отырып, мысалы, бағдарламалауға жүгінбей, тышқанның бірнеше шертулерімен мәліметтерді енгізуге болады. Бұл заманауи қолданбалы бағдарламалауда интерактивті технологияларды қолданудың көрнекі әдісі. Жүйе арқылы пайдаланушы интерфейсін бағдарламалауға тікелей байланысты бөлік визуалды бағдарламалау деп аталады.

Визуалды бағдарламалау болмаса, картаға түсіру процесі объектіні құратын код бөлігін жазуды қажет етеді. Кодталған нысандарды тек бағдарламаны орындау кезінде көруге болатын еді. Осы тәсілдің көмегімен нысандардың белгілі бір түрде көрінуі мен әрекет етуіне қол жеткізу бағдарламаның кодын бірнеше рет түзетуді, содан кейін бағдарламаны іске қосуды және соңында не болғанын бақылауды қажет ететін күрделі процесске айналады.

Көрнекі құралдардың көмегімен сіз нысандармен жұмыс жасай аласыз, оларды көздің алдында ұстап, нәтижеге бірден қол жеткізе аласыз. Объектілерді бағдарламаны орындау кезінде пайда болатындай етіп көру мүмкіндігі қолмен жасалатын көптеген құралдардың қажеттілігін жояды, бұл визуалды құралдары жоқ ортада жұмыс істейді - объектіге бағдарланған-болмағанымен. Нысан визуалды бағдарламалау ортасы түрінде орналастырылғаннан кейін, оның барлық атрибуттары бағдарламаның жұмысы кезінде орындалатын бірлік ретінде объектіге сәйкес келетін код түрінде бірден көрсетіледі.

Delphi-де объектілерді орналастыру объектілер мен нақты бағдарлама коды арасындағы тығыз байланысты. Нысандар формаға орналастырылады, ал сәйкес келетін код бастапқы файлға автоматты түрде жазылады. Бұл код визуалды ортаға қарағанда анағұрлым жоғары өнімділікті қамтамасыз ететін құрастырылған, тек бағдарламаны орындау кезінде ақпаратты түсіндіреді.

Пайдаланушыға ыңғайлы болу үшін графтармен жұмыс істеуге арналған визуалды редактор құру туралы шешім қабылданды. Бұл редактор ұшақтағы шеттер мен жиектердің суреттері түрінде график енгізуге мүмкіндік береді.

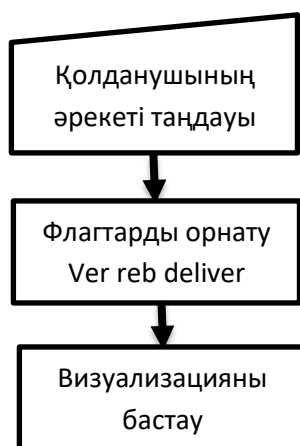


3.4 сурет – Бағдарламаның негізгі жұмыс үстелі

Мұндай редакторды құру Image компонентінің жұмысын қайта анықтауға қажет болды. Контекстік мәзірді көрсету үшін тінтуірдің оң жақ батырмасын басу керек.

Контекстік мәзірді көрсету үшін тінтуірдің оң жақ батырмасын басу керек.

```
procedure TForm1.Image1MouseDown(Sender: TObject;
  Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  if button <> mbLeft then // егер тышқанның сол жақ шертілген болса
    PopupMenu1.Popup(Mouse.CursorPos.X, Mouse.CursorPos.Y); // контекстік
    мәзірді көрсету
end;
```



3.5 сурет – Қабырғаларын немесе шыңын орналастыру үшін қолданылатын схема

Графикалық кескінді басқан кезде әрекетті таңдағаннан кейін, бұл әрекет графикалық құрылымды өзгерту арқылы көрсетіледі

```

1 procedure TForm1.Image1MouseDown(Sender: TObject;
2 Button: TMouseButton;
3 Shift: TShiftState;
4 X, Y: Integer);
5 var j,k,i,i2,xx,yy:integer;
6 begin If del_ver then begin // шыңды жою
7 //белгіленген шыңды табу
8 k:=findver(x, y, 10); // осы шыңнан шығатын барлық жиектерді алып тастаңыз
9 ver:=false;
10 reb:=false;
11 reb2:=false;
12 del_ver:=false;
13 if K=0 then exit;
14 for i:=1 to StringGrid3.ColCount-1 do if i>=k then vers[i]:=vers[i+1]; // ереп біз жойылған шыңның оң жағында болса,
15 біз бағандарды бір бірлікке кепі көшіреміз
16 for i:=1 to StringGrid3.RowCount-1 do for j:=1 to StringGrid3.RowCount-1 do if j>=k then StringGrid3.Cells[i,j]:=StringGrid3.cells[i,j+1];
17 for j:=1 to StringGrid3.RowCount-1 do for i:=1 to StringGrid3.RowCount-1 do if i>=k then StringGrid3.Cells[i,j]:=StringGrid3.cells[i+1,j];
18 pos:=pos-1;
19 edit1.Value:=edit1.Value-1;
20 StringGrid3.cells[0,0]:='*';
21 redraw;
22 Image1.Canvas.Pen.Mode:=pmCopy;
23 Button2.Caption:='Разместить вершину '+IntToStr(pos+1);
24 end;
25 If ver then begin inc(pos);
26 vers[pos].x:=x;
27 vers[pos].y:=y;
28 xx:=100;
29 yy:=100;
30 vers[pos].num:=pos;
31 ver:=false;
32 Button2.Caption:='Разместить вершину '+IntToStr(pos+1);
33 If pos>1 then begin StringGrid3.ColCount:=StringGrid3.ColCount+1;
34 StringGrid3.RowCount:=StringGrid3.RowCount+1;
35 Edit1.Value:=pos;
36 end;
37 for i:=1 to edit1.value+1 do begin
38 StringGrid3.Cells[i,0]:=inttostr(i);
39 StringGrid3.Cells[0,i]:=inttostr(i);
40 end;
41 for i:=1 to edit1.value do begin for i2:=1 to edit1.value do begin if StringGrid3.Cells[i2,i]='' then StringGrid3.cells[i2,i]:='0';
42 end;
43 end;
44 redraw; // шыңды қосқаннан кейін не болғанын суреттеу
45 end;
46 if reb2 then begin try reb:=false;
47 reb2:=false;
48 xx:=0;
49 yy:=0;
50 form1.Caption:=inttostr(findver(x, y));
51 i:=findver(x, y, 10);
52 If i <>0 then begin // ереп бәрі шыңның бір шеңберінен екінші шеңберге дұрыс көрсетілсе, онда көршілес матрицаға жазба қосыңыз
53 StringGrid3.Cells[findver(x, y, 10),
54 findver(xb, yb, 10)]:='1';
55 end;
56 StringGrid3.cells[0,0]:='*';
57 redraw;
58 except;
59 redraw;
60 end;
61 end;
62 if reb then begin try xb:=x;
63 yb:=y;
64 xx:=x;
65 yy:=y;
66 reb2:=true;
67 Image1.Canvas.MoveTo(0, 0);
68 Image1.Canvas.LineTo(x, y);
69 i:=findver(x, y, 10);
70 If i <>0 then begin inc(n);
71 StringGrid3.Cells[pos2,
72 0]:=inttostr(i);
73 i:=findver(x, y, 10); // енді қай қабырғаны қосатынымызды табамыз
74 end;
75 except;
76 reb:=false; // қабырғаны қосу сәтсіз болды
77 end;
78 end;

```

3.6 сурет – Бағдарламада графикалық кескінді басқан кезде әрекетті таңдағаннан кейінгі орындалатын функцияның Unit1.pas файлындағы коды

Енді пайдаланушы, мысалы, «добавления вершины» опциясын таңдап, тінтуірді графикалық кескін аймағына жылжыта алады. Шыңды орналастыру үшін таңдау орны экран бойымен жылжиды. Тінтуірдің сол жақ батырмасын шертіп, жоғарыда аталған процедураны орындағаннан кейін, шыңдар графикке қосылады.

Бағдарламадағы график іргелес матрица түрінде сақталады. Бұл матрица доғалардың салмағын сақтау үшін қолданылады. Бағандарда доғаның бағыты бар шыңдар, ал жолдарда қайдан.

Бағытталған график үшін матрица негізгі диагональға қатысты симметриялы болмауы мүмкін. Бағытталмаған график үшін ол әрқашан симметриялы болады.

StringGrid компоненті - бұл жолдардан тұратын кесте. Кесте деректерін тек оқуға немесе өңдеуге болады. Кестеде айналдыру жолақтары болуы мүмкін, ал бірінші жолдар мен бағандардың берілген саны бекітіліп, айналдырылмайды. Осылайша, құрамдас терезеде үнемі болатын баған мен жол тақырыптарын орнатуға болады. Кестенің әрбір ұяшығын қандай да бір объектімен байланыстыруға болады.

Көрсетілген мәтінді анықтайтын компоненттің негізгі қасиеттері:

- Cells[ACol, ARow: Integer]: string (ACol және ARow бағандары мен жолдарының индексі бар ұяшықтағы жол)
- Cols[Index: Integer]: TStrings (Index индексі бар бағандағы бағандар тізімі)
- Rows[Index: Integer]: TStrings (Index индексі бар бағандағы жолдар тізімі)

3.2 Графты сақтау және жүктеу

Стандартты файлды таңдау тілқатысу терезесін пайдаланып сақтау және жүктеу.

Dialogs компоненттер палитрасында Delphi OpenFileDialog компоненті және Delphi SaveDialog компоненті орналасқан. Осы қойындыда орналасқан барлық Delphi диалогтары, соның ішінде Delphi файлды таңдау диалогтары, визуалды емес, яғни. оларды жұмыс бағдарламасындағы Пішінге ауыстырған кезде олар көрінбейді, тек жобалау сатысында көрінеді. Delphi OpenFileDialog компоненті бағдарламада файлды ашық диалогтың стандартты Windows терезесін, Delphi SaveDialog компоненті - сақтау диалог терезесін ашуға мүмкіндік береді. Диалогтардан алынған мәліметтер белгілі бір графикалық файлмен жұмыс істеу үшін бағдарламаға жіберіледі.

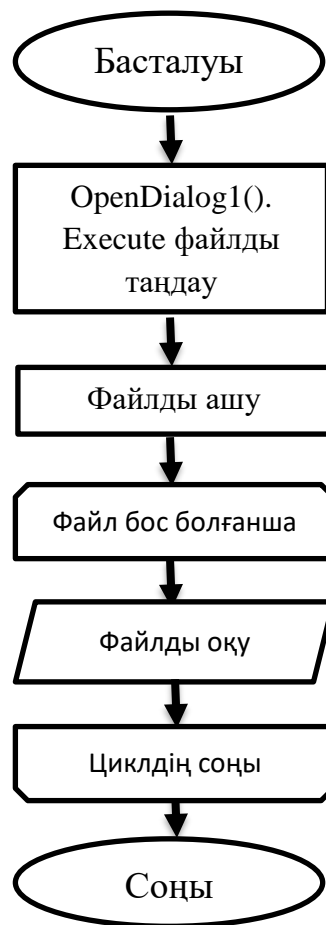
```

1 procedure TForm1.Button7Click(Sender: TObject);
2 var f: TextFile;
3 iTmp,i,k: Integer;
4 strTemp: String;
5 begin stringgrid2.Cells[0,0]:='';
6 stringgrid2.Cells[1,0]:='';
7 label2.Caption:='Макс. пр. ';
8 if (not OpenFileDialog1.Execute()) then exit;
9 StringGrid2.RowCount:=0;
10 AssignFile(f, OpenFileDialog1.FileName);
11 Reset(f);
12 with StringGrid3 do begin Readln(f, iTmp);
13 ColCount:=iTmp;
14 Readln(f, iTmp);
15 RowCount:=iTmp;
16 edit1.Value:=ColCount-1;
17 for i:=0 to ColCount - 1 do for k:=0 to RowCount - 1 do begin Readln(f, strTemp);
18 Cells[i,k]:=strTemp;
19 end;
20 end;
21 CloseFile(f);
22 if StringGrid3.ColCount>10 then button6.Click else button5.Click;
23 Button2.Caption:='Разместить вершину '+IntToStr(pos+1);
24 end;

```

3.8 сурет – Графты жүктеу функциясының Unit1.pas файлындағы коды

Жүктеу кезінде файл таңдалады, жазу үшін ашылады. Одан формадағы Stringgrid компонентімен ұсынылатын іргелес матрицаны оқиды.

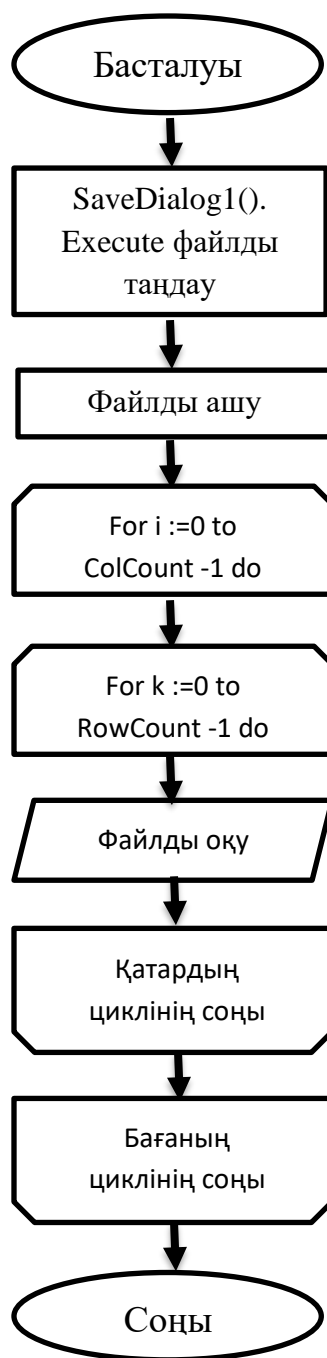


3.9 сурет – Файлды жүктеу схемасы

Файлды сақтау кезінде ұқсас принцип қолданылады. Файл таңдалады, жазу үшін ашылады және көршілес матрицаның бағандары мен жолдарының саны туралы ақпарат сақталады, содан кейін матрицаның өзі.

```
1 procedure TForm1.Button8Click(Sender: TObject);
2
3 var
4     f: TextFile;
5     i, k: Integer;
6 begin
7     if (not SaveDialog1.Execute()) then exit;
8     AssignFile(f, SaveDialog1.FileName);
9     Rewrite(f);
10    with StringGrid3 do
11        begin
12
13            Writeln(f, ColCount);
14            Writeln(f, RowCount);
15
16            for i: = 0 to ColCount - 1 do
17                for k: = 0 to RowCount - 1 do
18                    Writeln(f, Cells[i, k]);
19                end;
20            CloseFile(f);
21        end;
```

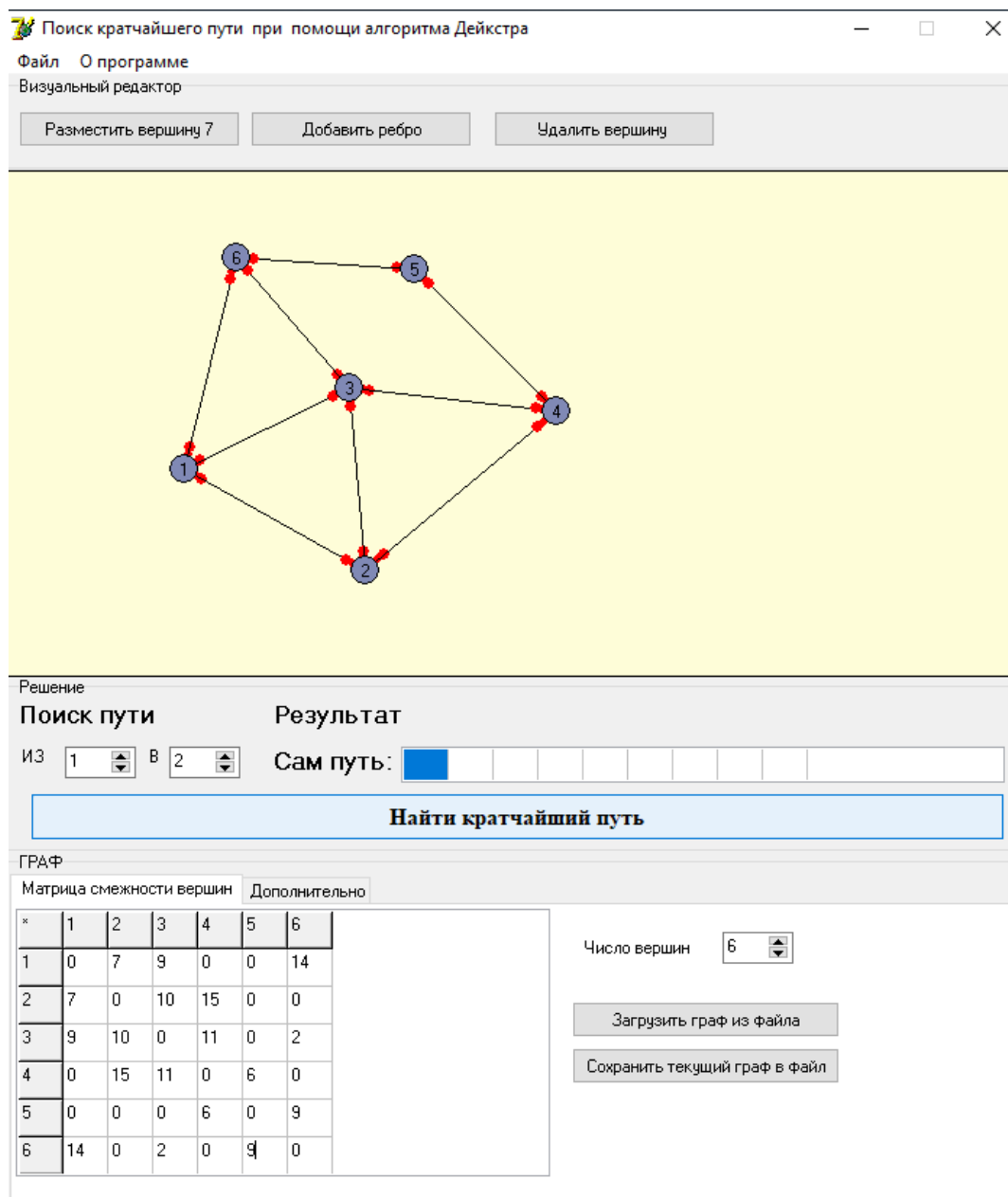
3.10 сурет – Unit1.pas файлындағы Графты сақтау функциясының коды



3.11 сурет – Файлды сақтау схемасы

Бұл мәселені шешкен кезде ақпараттық материалдарды ұсыну үшін Delphi конструкторының тілін қолдану оңтайлы болып табылады, бұл интерфейстің жоғары деңгейлі құралы болып табылады және қосымшаларды тез әрі тиімді жасауға мүмкіндік береді. Пайдаланушымен өзара әрекеттесу экран формалары арқылы жүзеге асырылады.

Төменде негізгі терезенің пішінінің мысалы келтірілген.



3.12 сурет – Негізгі терезе

Енді негізгі терезеде жалпы берілгендердің ішінде көзге түсетіні жұмыс жасау үстелі мен шыңдардың матрицасы.

3.3 Дейкстрдің сейрек графикалық алгоритмін қайта есепке ала отырып, анализ жасау

Болашақта бағдарламаны және жазбаша кодты қолдану мүмкіндігін түсінуді жеңілдету үшін Дейкстр алгоритмін жүзеге асыру үшін модуль әзірлеу туралы шешім қабылданды. Бұл модульдің жұмысы жоғарыда аталған алгоритмге толық сәйкес келеді.

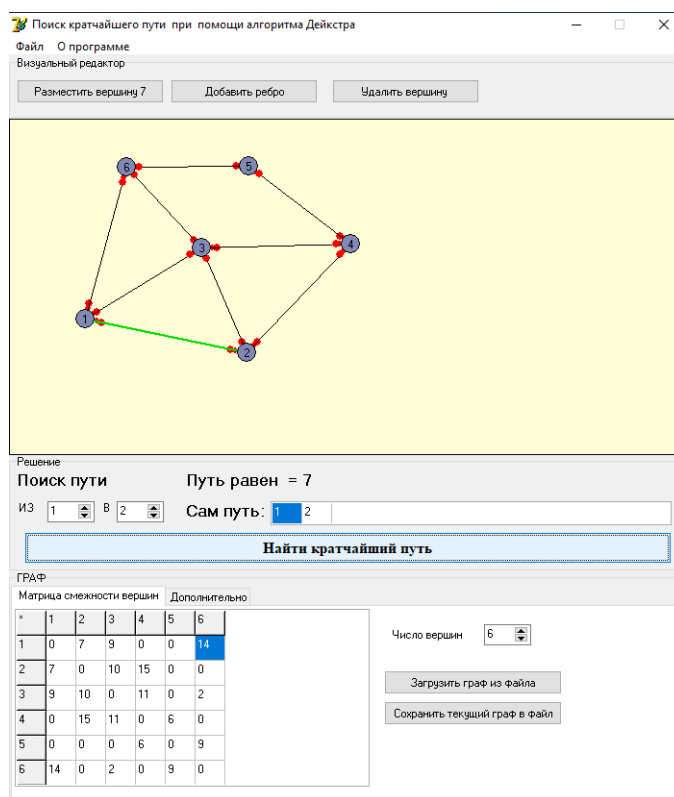
```

1 unit deijkstra;
2 interface const max=100;
3 inf=100*100*100;
4 Type TJarak=Array [1..max,
5 1..max] of Integer;
6 TPath=record nodeke: byte;
7 Arraypath: Array [1..max] of byte;
8 Jarak: Integer;
9 end;
10 Procedure RuteTerpendek(var Data: TJarak;
11 var Closed: TPath;
12 var awal, tujuan: integer;
13 count: byte);
14 Implementation Procedure RuteTerpendek(var Data: TJarak;
15 var Closed: TPath;
16 var awal, tujuan: integer;
17 count: byte);
18 Var Path: Array [ 1..max] of TPath;
19 Open,
20 Jauh: Array[1..max] of Real;
21 procedure Initdata(count: byte);
22 var i,
23 j: byte;
24 begin fillchar(Open, sizeof(Open), inf);
25 fillchar(jauh, sizeof(jauh), inf);
26 fillchar(Path, sizeof(Path), inf);
27 for i:=1 to count do begin open[i]:=0;
28 path[i].Jarak:=0;
29 jauh[i]:=data[awal,
30 i];
31 path[i].arraypath[1]:=awal;
32 path[i].nodeke:=1;
33 path[i].arraypath[2]:=i;
34 path[i].nodeke:=path[i].nodeke + 1;
35 path[i].Jarak:=Data[path[i].arraypath[1],
36 path[i].arraypath[2]] end;
37 open[awal]:=awal;
38 end;
39 procedure update_close(var Closed: TPath);
40 begin closed:=path[tujuan];
41 end;
42 function successor(count: byte): byte;
43 var i,
44 j: byte;
45 minimum: Real;
46 begin minimum:=inf;
47 for i:=1 to count do if (jauh[i] < minimum) and (open[i]=0) then begin minimum:=jauh[i];
48 j:=i;
49 end;
50 successor:=j;
51 end;
52 procedure lessthen(count, x: byte);
53 var i: byte;
54 a,
55 b: Real;
56 begin for i:=1 to count do begin a:=jauh[x] + data[x,
57 i];
58 b:=jauh[i];
59 if a < b then begin jauh[i]:=a;
60 path[i]:=path[x];
61 path[i].nodeke:=path[i].nodeke + 1;
62 path[i].arraypath[path[i].nodeke]:=i;
63 path[i].Jarak:=path[i].Jarak + Data[path[i].arraypath[path[i].nodeke-1],
64 path[i].arraypath[path[i].nodeke]] end;
65 end;
66 end;
67 procedure disktra(count: byte);
68 var i,
69 j: byte;
70 begin for i:=1 to count do begin lessthen(count, successor(count));
71 update_close(closed);
72 open[successor(count)]:=successor(count);
73 end;
74 end;
75 begin initdata(count);
76 disktra(count);
77 end;
78 end.

```

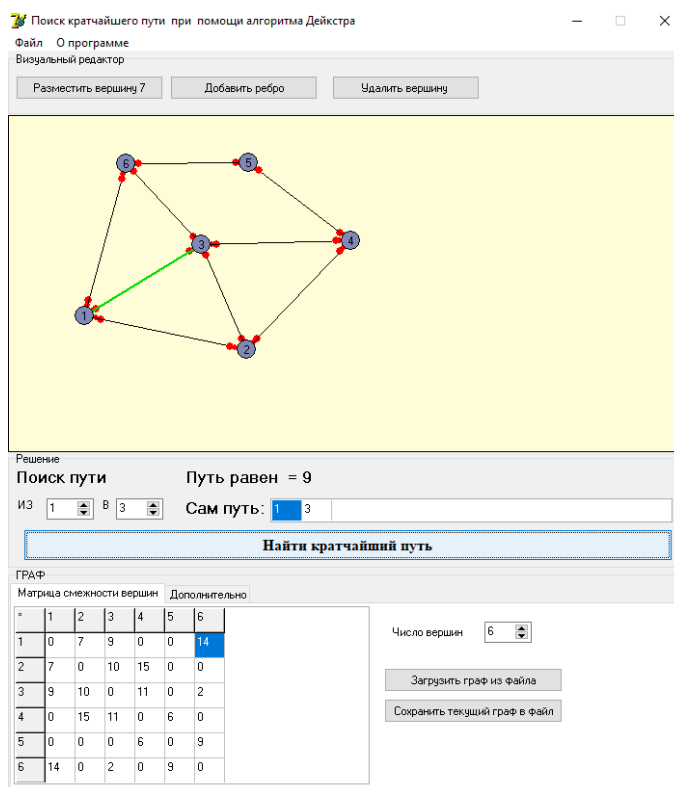
3.13 сурет – Unit1.pas файлындағы негізгі модульдің коды

Енді 2.4 бөліміндегі мысалды осы бағдарлама көмегімен жасайық. Ол үшін берілген барлық алты шыңды орнатып, сәйкесінше әр қайсысының арасындағы ара қашықтықтарды беріп шығамыз. Содан кейін жолды іздеу үшін қай жолдан қай жолға екенін шыңның реттік номерін көрсетеміз.

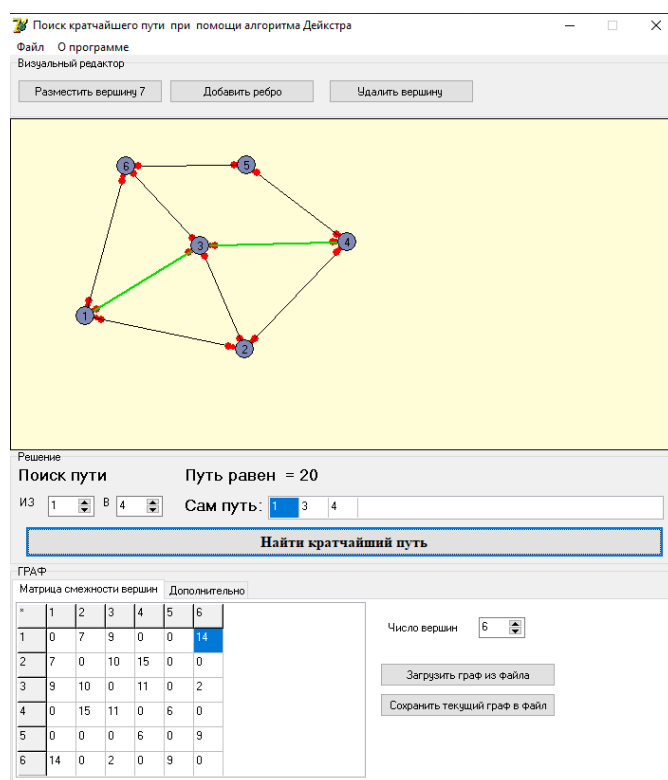


3.14 сурет – Бірінші шыңнан екінші шыңға дейінгі ең қысқа арақашықтық

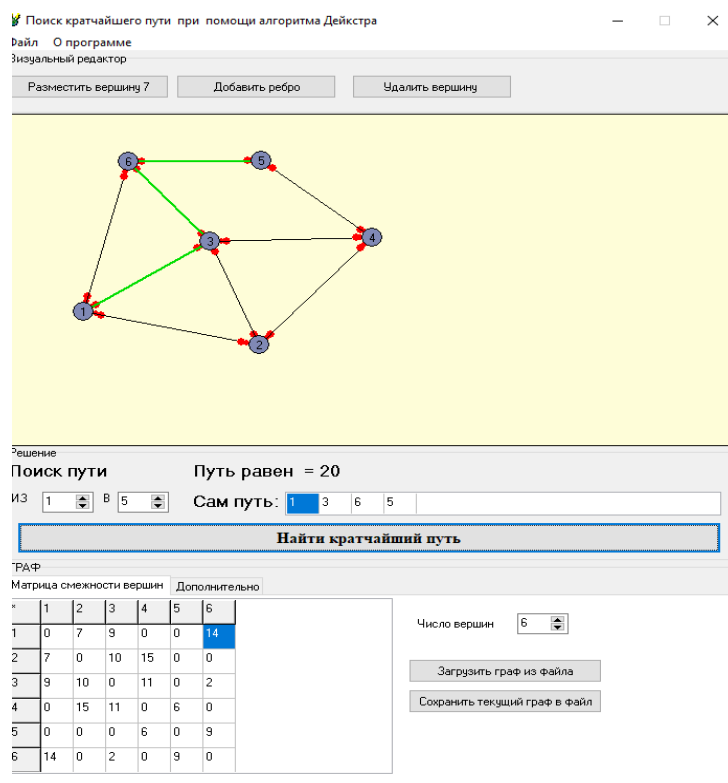
Сонымен бірінші шыңнан екінші шыңға дейінгі ең қысқа арақашықтық теориядағы мысалмен сәйкес келеді. Әрі қарай жалғастырып басқа шыңдар арасындағы ең қысқа ара қашықтықты іздейік.



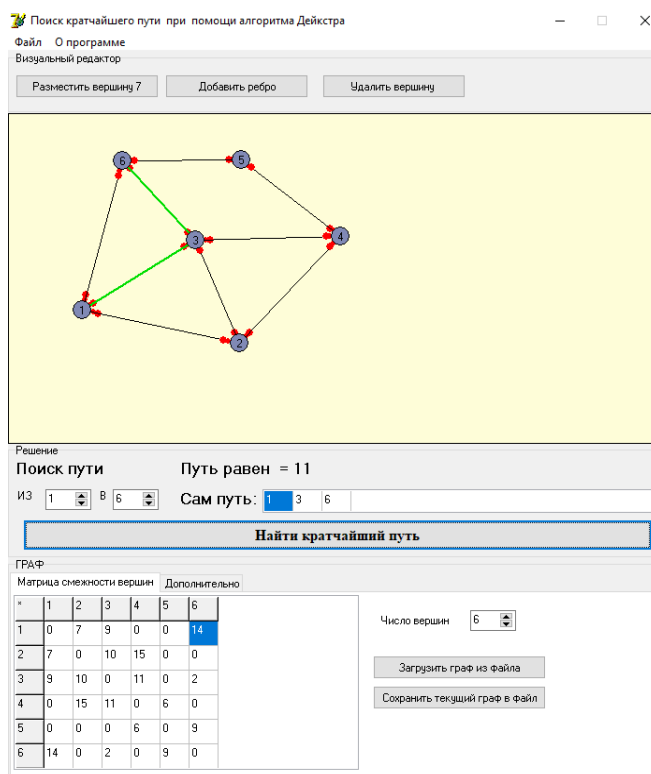
3.15 сурет – Бірінші шыңнан үшінші шыңға дейінгі ең қысқа арақашықтық



3.16 сурет – Бірінші шыңнан төртінші шыңға дейінгі ең қысқа арақашықтық



3.17 сурет – Бірінші шыңнан бесінші шыңға дейінгі ең қысқа арақашықтық



3.18 сурет – Бірінші шыңнан алтыншы шыңға дейінгі ең қысқа арақашықтық

Қорытындылай келе алгоритмнің нәтижесі былай болды: жоғарғы 1-ден 2-ге дейінгі қысқа жол - 7, 3 - 9, 4 - 20, 5 - 20, 6 – 11. Сонда алгоритм толықтай

дұрыс жұмыс жасап тұр. Өйткені, теориялық тұрғыда есептелінген нәтиже бағдарлама жасаған нәтижемен сәйкес келді.

4 Өміртіршілік қауіпсіздігі

4.1 Еңбек шартын талдау

Бұл дипломдық жұмыста Дейкстр алгоритмін қарастырылды және маршрутизаторлар арасындағы ең тиімді қашықтықты есептедік. Бұл жоба электрондық құрылғыда, яғни компьютерде, адам ағзасына әсер ететін жерде жүзеге асырылады. Тіршілік әрекетінің қауіпсіздігінің осы бөлігінің мақсаты - электронды құрылғылардан келетін зиянды азайту. Адам баласы күн сайын өзінің өмірлік қажеттіліктерін қанағаттандыру үшін жұмыс істейді. Белгілі бір жұмыстарды орындау, бұл, адам өмірі мен денсаулығына үнемі қауіп төндіретін, төтенше жағдайлардың пайда болуына ықпал етеді.

Адам өмірі мен денсаулығына қауіп төнген жағдайда мінез-құлқының күрделенуі, сондай-ақ, келесі факторларға байланысты төтенше жағдайлардан қауіпсіз құтылуы:

- уақытылы тәуекелге бел бууы;
- дұрыс диагноз қойып, қауіп-қатерге тиісті түрде әрекет ету тәсілдерін таңдау.

Кеңседе қызметкерлерге арнап орын бөлінген. Қызметкерлердің жұмысы келіп түсетін ағындарды өңдеу, талдау және анализ жүргізу болып табылады. Мұның бәрі дербес компьютерлерге орнатылған арнайы бағдарламалар арқылы жүзеге асырылады, сол себептен қызметкер көп уақытын компьютердің алдында өткізеді.

Қызметкерлер саны 2 адамнан тұрады: әзірлеуші – 1; жобаның менеджері – 1. Жұмыстың ауырлығы мен қарқындылығына сәйкес, жұмыс І санатқа жатады – қолайлы жүктемелермен оңтайлы жұмыс жағдайында орындалады.

4.1.1 Жасанды жарықтандыру

Жасанды жарықтандыру – бөлмелерде немесе өндірістік мекемелерде табиғи жарықтың аз мөлшерде болуына байланысты немесе оның мүлде болмауы кезінде қолданылатын жарықтандырудың бір түрі. Ол орындалу конструкциясына қарай келесідей түрлерге бөлінеді [1]:

- жалпы жарық – өндірістік мекеменің, не болмаса, бөлменің барлық ауданын бір қалыпты жарықтандыруға негізделген;
- жергілікті жарық – жұмыс істелінетін орындардың әрқайсын жеке дара жарықтандыруға негізделген;
- қосынды жарық – өндірістік мекемеде немесе бөлмеде, жалпы жарықта, жергілікті жарықта бірдей қолданылатын, жасанды жарықтандырудың бір түрі. Қосынды жарық тиімді болу себебінен өндірістерде кеңінен пайдаланылуда.

Неге арналғанына байланысты жасанды жарықтандырудың түрлері келесідей болады:

– жұмыс жарығы – өндіріс мекемелерінің бөлмелерінде күнделікті орындалатын жұмыстарды ойдағыдай жасау үшін арналған жарықты қамтамасыз етуге арналған;

– авариялық (апат) жарық – жұмыс жарығы күтпеген жағдайда сөніп қалған кездерде, жұмысты тоқтатпай, ары қарай жалғастыра беру үшін негізделген. Сондықтан да, жарықтың бұл түрінде өзіндік, бөлек бір, электр ток көзі болған дұрыс;

– эвакуациялық жарық – жұмыс жарығы кенет сөніп қалған кезде, бөлмелерден, ғимараттардан адамдарды қауіп-қатерсіз, аман-есен шығару мақсатына негізделген;

– кезекші жарық – жұмыс істелінбейтін кездерде, мезгілдерде бөлмелерді жарықпен қамтамасыз етуге арналған;

– күзетші жарық – жұмыс уақыты толығымен аяқталғанда, түнде территорияны (аланды) жарықпен қамтамасыз етуге негізделген.

Жүргізілген барлық жұмыс көзге ауыртпалық түсіреді. Осыған байланысты ҚНЖЕ ҚР 2.04-104-2012 [2] «Табиғи және жасанды жарықтандыру. Проектілеу нормалары» мен орнатылған жарықтандыруды сәйкестендіру жүргізіледі.

Жасанды жарықты есептеу. Құрамында екі шамнан тұратын шамшырақтарды қолдану люминесценттік шамдарға тән стробоскоптық әсерден құтылу мақсатында қолдандық. Сол үшін оларды кері фазада қосады. Мұндай шара өндірістік жарақаттарды азайтуға қызмет етеді. Жаппай жарықтандыру жүйесін КТШ (күндізгі түс шамдары) типті люминесцентті шамдармен қолданамыз

Сәйкесінше, өміртіршілік қауіпсіздігі бөлімінде жұмысшыға зиянын тигізуі қауіпі болатын келесі факторларды қарастыра отырып, олардың алдын алу мақсатында есептеулер жүргізілді:

Жұмысшы өзінің жұмыс орнында келесідей зиянды факторларға тап болады:

– құрамы мен қысымы, салыстырмалы ылғалдылығы, ауа ағынының жылдамдығы ауа температурасымен анықталатын микроклиматтық жағдайлары;

– өндірістік шудың әсері;

– психофизиологиялық факторлар, яғни, көру және есту анализаторларына ауыртпашылық қысымның мөлшерден көп келтірілуі, жұмыстың біркелкілігі, эмоционалды ауыртпашылықтар және т.б.;

– адамды электр тогының соғуы;

– өрттің пайда болуы.

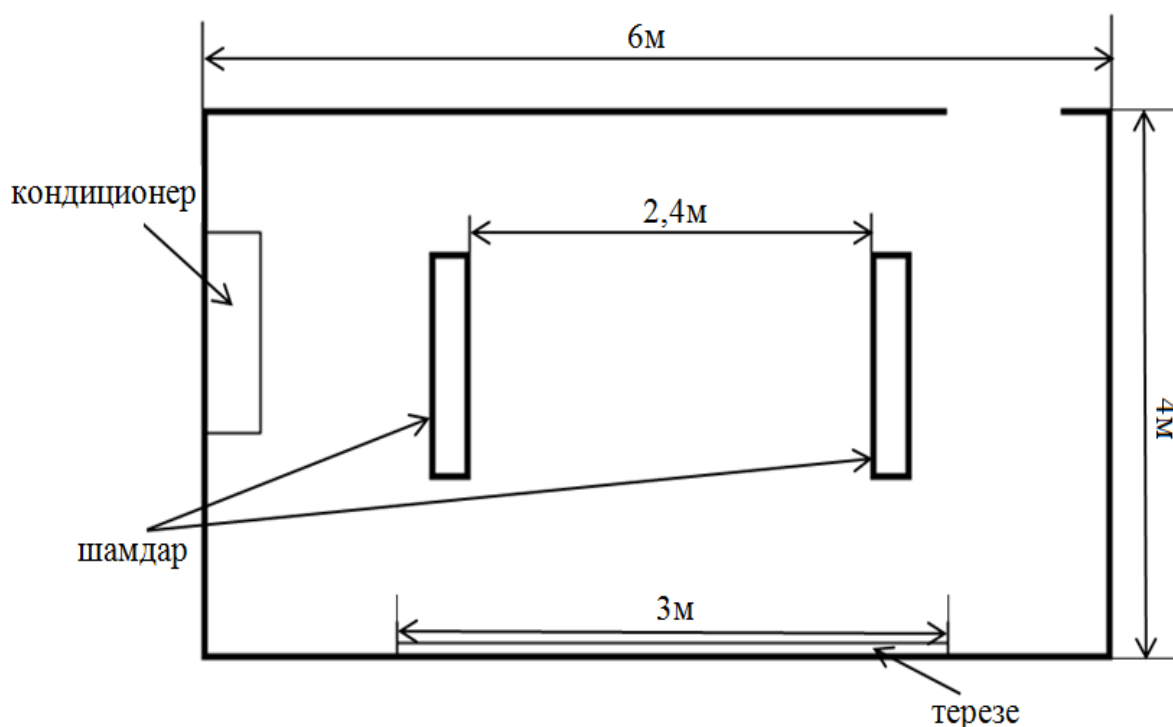
Өміртіршілік қауіпсіздігі бөлімінде жұмысшыға зиянын тигізетін келесі факторларды қарастыра отырып, олардың алдын алу мақсатында есептеулер жүргіземін:

а) Жасанды жарықтандыру жүйесі;

ә) Кондиционерлеу жүйесі.

Жабдықтардың орналасу жоспары 4.1-суретте көрсетілген. Бөлменің ұзындығы $A=6\text{м}$, ені $B=4\text{м}$, биіктігі $H=3\text{м}$. Терезе саны 1, биіктігі 2 м, ені 3 м.

Операторлық кеңседе 2 адам болады, бір адамға 12 м^2 бөлме ауданынан кем емес орын беріледі. Бөлменің ауданы 24 м^2 .



4.1 сурет – Бөлмедегі терезе мен шамдардың орналасу схемасы

4.1.2 Жұмыс орнындағы микроклимат параметрлері

Оператордың жұмыс жағдайы көбінесе операторлық кеңсенің микроклиматына байланысты. МЕМСТ 12.1.005-88 «Жұмыс аймағында ауа, жалпы санитарлық-гигиеналық талаптар» [3] литерат нормаларына сәйкес орындалатын жұмыстар жеңіл физикалық жұмыстарға жатады және жұмыс санатының 1б санатына сәйкес келеді. Мұндай жұмыс жағдайы отырып, тұрып, жүріп және де кейбір физикалық күш түсіру арқылы орындалады. 1б санаты үшін микроклимат параметрлерінің оңтайлы нормаларына сәйкес, суық мезгілде бөлмедегі ауа температурасы $21-23^{\circ}\text{C}$ аралығында, жылы кезеңде температура $23-25^{\circ}\text{C}$ аралығында және ауаның салыстырмалы ылғалдылығы 75/60% астам болды. Ол үшін қыс мезгілінде – жылыту, жаз мезгілінде – салқындату жүйелері қолданылады. Бөлмедегі оңтайлы микроклимат салқындату жүйесі арқылы іске асырылады. Операторлық кеңсе үшін оңтайлы және жарамды температура мен салыстырмалы ылғалдылықтың нормалары, жұмыс санаты мен жыл мезгілін ескере отырып алғандағы ауа қозғалысының жылдамдығы 4.1-кестеде келтірілген.

Бөлмеде тұрақты шу мен діріл көздері жоқ. Шудың сыртқы көздерінен бөлме дыбыс өткізбейтін пластикалық терезелермен қорғалған.

4.1 кесте – Өндірістік үй-жайлардың аймағындағы температураның, салыстырмалы ылғалдылықтың және ауа қозғалысының жылдамдығының нормаланған өлшемдері

Жыл мезгілі	Температура, °С			Салыстырмалы ылғалдылық, %		Ауа жылдамдығы, м/с	
	Оңтайлы	Жарамды		Оңтайлы	Жарамды	Оңтайлы	Жарамды
		Төменгі шек	Жоғарғы шек				
Суық	22-24	25	21	40-60	75	0.1	0.1 – ден артық емес
Жылы	23-25	28	22	40-60	28	0.1	0.1-0.2

4.1.3 Компьютер операторының жұмыс орнына қойылатын жалпы гигиеналық талаптарды сипаттау

Компьютермен жұмыс істеу кезінде температура жоғарылап, ауаның салыстырмалы ылғалдылығы азайып, ауаның құрамы өзгертінді белгілі. Сондықтан кабинетте төмендегідей микроклиматтың үйлесімді параметрлерін ұстаған жөн.

Видеомонитор экранына қарсы қабырғаны қара, қоңырқай түстермен бояуға болмайды. Жұмыс столының бетінің түсі табиғи ағаш түсі немесе ашық жасыл, ашық көк болғаны дұрыс.

Жұмыс столының ең төменгі жарықталынуы 400-500 люкс болды. Ауаның температурасы – 19-21⁰С, салыстырмалы ылғалдылығы - 62-55%, ауаның қозғалыс жылдамдығы – 0.1 м/с. Бөлмедегі микроклиматты сақтау үшін күн радиациясы тіке түспейтін жаққа (солтүстік солтүстік шығыс) орналастырылды. Ауаның ылғалдылығын жоғарылату үшін қыздырғыш құралдардың тұсына үлкен көлемді аквариумдарды орналастыруға болады.

Жоғарыда аталған микроклиматты ұстап тұру үшін бөлменің ауасын есік-терезе арқылы желдетіп тұрдық. Желдету уақыты 10 минуттан кем емес. Ауаны тазартудың басқа жолы терезеге орналастырылған кондиционер арқылы ауа алмастыруға болады. Ауа алмасу бөлменің ауданы мен квадратурасына байланысты.

4.1.4 Компьютерден алынатын электромагниттік сәулелену

Компьютерлер бүкіл әлемдегі миллиондаған адамдардың күнделікті қызметінде пайдаланады. Біздің елде компьютерлендіру ауқымды болып табылады және көптеген жүздеген мың адам жұмыс күнінің көп бөлігін және бос уақытты экранның алдында өткізеді.

Адам денсаулығына әсерлі қауіпті ұялы телефондар, микротолқынды пештер, компьютерлер мен теледидарлар тигізеді. Микротолқынды пештер, негізінен, қысқа уақыт ішінде жұмыс істейді (орта есеппен 1-ден 7 минутқа

дейін), теледидарлар көрермендерден жақын қашықтықта орналасқанда ғана үлкен зиян келтіреді.

Бірақ біз ұзақ уақыт бойы өзара әрекеттесетін электромагниттік сәулелену көздері сәйкесінше ең қауіпті болып табылады. Осы тұрғыдан алғанда, компьютердің электромагниттік сәулелену мәселесі, яғни компьютердің адам ағзасына әсер етуі бірнеше себептерге байланысты өте өткір болады:

- Компьютерде бірден екі электромагниттік сәулелену көзі болады (монитор мен жүйелік блок)

- ДК пайдаланушысы қауіпсіз қашықтықта жұмыс істеу қабілетінен алыстау болып келеді

- Компьютердің ұзақ уақытқа әсері (қазіргі заманғы пайдаланушылар үшін компьютерде күніне 6 сағаттан артық жұмыс істеуге тыйым салатын ресми нормалармен 12 сағаттан артық).

Сонымен қатар, жағдайды нашарлататын бірнеше қайталама факторлар бар, олардың ішінде тығыз үй-жайда жұмыс істемейді және көптеген компьютерлердің бір жерде концентрациясы бар.

Ноутбуктің немесе компьютердің сәулеленуі қолданып отырған адамға ғана емес, сонымен қатар жанында тұрған адамдарыңызға да әсер етеді!

Компьютерде отырған адамнан басқа, ең осал адам сіздердің алдыңызда отыратын/тұрып тұрған немесе оң жақ бұрышта. Әрине, қашықтықтың да рөлі маңызды 1.5 м дейін аймақ қауіпті деп саналады.

Бірақ әрбір технологиялық бірліктің электромагниттік сәулеленуі бөлменің жалпы электромагниттік фонын арттырады.

Компьютердің зиянды әсері қандай?

Электромагниттік өрістің көбеюі компьютердің адам денсаулығына кері әсерін қамтамасыз етеді.

Бірнеше күн бойы компьютерде жұмысын жалғастырудың нәтижесінде адам өзін әлсіз сезіне бастайды, өте тітіркендіргіш, сұрақтарға бірінғай жауап береді, бірден жауап береді, ұйқықұмарлығы көбейеді. Қазіргі қоғамдағы бұл құбылыс созылмалы шаршау синдромы деп аталды және ресми медицина бойынша емделуге болмайды.

Қорғау жабдығын пайдаланбай компьютермен үнемі жұмыс істеудің салдары [4]:

- Пайдаланушылардың 60% – көру органдарының аурулары;
- Пайдаланушылардың 60% – жүрек-қантамыр жүйесі аурулары;
- Пайдаланушылардың 40% – асқазан-ішек жолдарының аурулары;
- Пайдаланушылардың 10% – тері аурулары;
- түрлі ісіктер.

Сонымен компьютердің адам ағзасына залалын тигізетін факторлары мынадай:

- көздің көру қабілетін төмендетеді;
- омыртқалардың қисаюына әкеледі;
- жүйкеге салмақ түсіреді;

– шаршағыштық, әлсіздік басады.

Компьютермен жұмыс істеу ережелері:

– компьютердің мониторы мен көзіңіздің арақашықтығы кемінде 50 см болсын; компьютер тұрған бөлменің ауасы жиі тазартылуы керек;

– күн сайын бөлмеде ылғалды тазалық жұмыстарын (влажная уборка) жүргізгеніңіз дұрыс;

– компьютермен жұмыс істеп болғаннан кейін қолыңызды салқын сумен жуыңыз;

– көзге жасалатын жаттығуларды жасап отыру керек;

– оқуға тиісті құжаттарды принтерден басып шығарып алып оқу тиіс, көзіңіздің саулығына үлкен пайдасы тиеді.

4.1.5 Өрт қауіпсіздігі

Өрт уақытымен және кеңістікте көбейетін, жанудан бөлек масса және жылу ауысуын қосатын, ауыр физико-химиялық процесс. Оның алдын алу үшін төмендегі құралдардың болуы талап етілсін.

Өрттерді сөндірудің негізгі құралдары:

Су — өртті сөндірудің негізгі құралы. Су электр өткізгіш, сондықтан оны тоқта тұрған жүйелер мен жабдықтарды сөндіруге пайдалануға болмайды. Электр сымдарға судың түсуінен қысқа тұйықталуы болуы мүмкін.

Құм — өрттің шағын ошақтарын сөндіру құралдарының ең қапайымы болып табылады. Оның отты сөндірудегі әрекеті — ол жанғаш затты салқындатады

Химиялық көбіктің құндылығы оның төзімділігінде, тұтқырлығында, жеңілдігінде болып табылады, ол жанған затты толығымен тығыз жауады және оған ауа жібермейді. Көмір қышқылды өрт сөндіргіштер химиялықтармен қатар айтарлықтай таралған. Олар химиялық зертханаларда, кітапханаларда, мұражайларда және басқа да жайларда кеңінен қолданылады. Бұл өрт сөндіргішті қуаттандыру үшін үлкен қысымда тұрған сұйытылған көмір қышқылы (көміртек диоксиді) қолданылады. Қазіргі уақытта құрғақ ұнтақтармен толтырылған ұнтақты өрт сөндіргіштер көп таралған, олар өрт аумағына түсіп, отты сөндіретін зат бөледі. Ұнтақты өрт сөндіргіштер тез тұтанғыш және жанғыш сұйықтықтардың, газдардың, ағаш өнімдерінің, тоқ қысымында тұрған электр сымдардың, сілтілі металлдар мен өздігінен жану қасиеті бар басқа да заттардың жануын сөндіру үшін қолданылады [5].

4.2 Өміртіршілік қауіпсіздігінің техникалық шешімі

4.2.1 Кондиционерлеу жүйесін есептеу

Бұл есептеу «Әуе желілерін есептеу және желдеткіштерді таңдау» әдістемелік нұсқаулық көмегімен жүзеге асырылды [6].

Жұмыс орнында қалыпты микроклимат пен тазалықты қамтамасыз ету үшін кондиционерлеу қолданылады, яғни желдетуші қондырғылар көмегімен бөлмеде жасанды микроклимат құрылады. Жасанды микроклиматты құру

кезінде бөлме ішіне МЕМСТ 12.1. 005-88 ЕҚСЖ [2] ұсынысы бойынша сырттан таза ауа жіберілді.

Кондиционерлеу жұмыс орнындағы микроклиматты, жұмыс жағдайының жақсаруы мен аппараттың сезімталдығын қамтамасыз етеді.

Едәуір жылу бөлінетін ғимараттардағы кіру ауасының көлемі $m^3/сағ$, жылуды жұту формуласы:

$$L = \frac{\sum Q_a}{C \cdot \gamma \cdot (t_{aui} - t_{kip})} (m^3/сағ) , \quad (4.8)$$

мұндағы Q_a - айқын жылылықтың бөлінуі, Дж/сағ;

C – ауаның жылулық сыйымдылығы, Дж/(кг·град);

t_{aui} – қашықтайтын бөлме ауасының температурасы, $30^\circ C$;

t_{kip} – кіретін ауаның температурасы, $22^\circ C$;

γ – кіретін ауаның тығыздығы ($1.4 \text{ кг}/m^3$).

Айқын жылылықтың бөліну формуласы:

$$Q_a = Q_1 + Q_2 + Q_3 + Q_4 , \quad (4.9)$$

мұндағы Q_1 – аппаратурадан жылу бөліну;

Q_2 – жарықтану көзінен жылу бөліну;

Q_3 – адамдардан жылу бөліну;

Q_4 – терезеден күн радиациясының түсуі.

Аппаратурадан жылу бөліну формуласы:

$$Q_1 = \eta_1 + \eta_2 + \eta_3 + \eta_4 + N_{ном} , \quad (4.10)$$

мұндағы η_1 – қондыратын қуаттың қолдану коэффициенті, ол 0.7-0.9 тең;

η_2 – жүктеме коэффициенті, ол 0.5-0.8 тең;

η_3 – аппаратураның бір уақытта жұмыс істеу коэффициенті, ол 0.5-1 тең;

η_4 – жылы энергияға өткен кездегі бөлме ішіндегі ауа жылуының ассимилялық коэффициенті, ол 0.65-1.0 тең;

$N_{ном}$ – барлық аппаратураның номиналды қуаты.

$$Q_1 = 0.8 + 0.7 + 0.7 + 0.8 + 90 = 93 \text{ (Дж/сағ)}$$

Жалған жарықтандыру көздерінен бөлінген жылуды есептеу, практикалық жүзінде қолданылған барлық энергия, қорытындысында жылуға айналады:

$$Q_2 = \phi \cdot N_{жар} , \quad (4.11)$$

мұндағы ϕ – жылуға айналатын энергияның мөлшерін қарастыратын коэффициенті, $\phi=0,8$;

$N_{жар}$ – бөлмедегі жарықтану құрылғысының қуаты (әрқайсысы 120 Вт, 4 шам).

$$Q_2 = 0.8 \cdot 2 \cdot 30 = 48 \text{ (Дж/сағ)}$$

Адамдардан жылу бөліну мына формуламен есептеледі:

$$Q_3 = n \cdot q_n , \quad (4.12)$$

мұндағы n – жұмысшы саны;

q_n – бір адамның жылу шығаруы, 80-116 Вт.

$$Q_3 = 1 \cdot 100 = 100 \text{ (Дж/сағ)}$$

Әйнектен күн радиациясының түсуі арқылы болатын жылу мөлшерін есептеу:

$$Q_4 = F_{ayd} \cdot q \cdot m \cdot k , \quad (4.13)$$

мұндағы F_{ayd} – терезе ауданы, м²;

m – терезе саны;

k – металлды түптеу үшін түзеткіш еселік, $k = 1.25$;

q – 1 м² терезе арқылы жылу бөліну, $q = 224 \text{ Вт/м}^2$.

$$Q_4 = 6 \cdot 1 \cdot 1.25 \cdot 224 = 1680 \text{ (Дж/сағ)}$$

Анық бөлінетін жылуды (4.9) формуласы бойынша анықтаймыз:

$$Q = 93 + 48 + 100 + 1680 = 1921 \text{ (Дж/сағ)}$$

Бөлме ішіндегі артық жылулықты жұту үшін, керек ауаның мөлшері:

$$L = \frac{1921}{1 \cdot 1.4 \cdot (30 - 22)} = 171.5 \text{ (м}^3\text{/сағ)}.$$

4.3 кесте – Кондиционердің негізгі сипаттамалары.

Жалпы аталуы	Samsung AR12MSFPAWQNERWhite
Түрі	қабырғалық сплит жүйесі
Ауданы, (м ²)	21
Негізгі режимдер	Салқындату/жылыту
Инвертор	Бар
Бақылануы	электрондық, қашықтан басқару пульті
Салқындату қуаты, (Вт)	2200
Жылыту қуаты, (Вт)	2375
Салқындату кезінде тұтынатын электр қуаты, (Вт)	685
Жылыту кезіндегі тұтынатын электр қуаты, (Вт)	659
Ұсынылатын жылыту/салқындату ауданы, (м ²)	21
Ішкі блоктың өлшемдері, (см)	730x254x170 мм
Сыртқы блоктың өлшемдері, (см)	730x254x170 мм
Ішкі блоктың салмағы	8 кг
Сыртқы блоктың салмағы	23.5 кг

Алынған ақпараттар бойынша, ылғалдылықты, тазалықты, ауаның қозғалу жылдамдығын және температураны автоматты түрде ұстап тұратын, желдеткіш ретінде Samsung AR12MSFPAWQNERWhite кондиционерін таңдадық. Кондиционердің негізгі сипаттамалары 4.3-кестеде келтірілген.

4.2.2 Компьютерден алынатын электромагниттік сәулелену

Қорғайтын құрылғыны ЭМӨ әсерінен есептеу әдістемесі материалдар мен конструкцияларды қолдану тиімділігін бағалау болып табылады.

Бастапқы есептеу мәліметтері экранның және технологиялық саңылаулардың геометриялық өлшемдері, экран материалдарының электрлік және магниттік сипаттамалары, радиациялық толқын ұзындығы, жұмыс аймағындағы өріс күші және ЭҚК-де болу ұзақтығы.

Есептеу тәртібі келесідей:

Электромагниттік өрістің көзінен қай жерде (жақын немесе алыс) есептелетін нүкте (жұмыс орны) орналасқандығы және электромагниттік өрістің қарқындылығы қандай параметрлермен есептелетіні анықталады.

4.4 кесте - ЭҚК әсер ету аймақтары

$r \ll \lambda/2\pi$ жақын аймақ	Аралық аймақ	$r \ll \lambda/2\pi$ алыс аймақ
ЭҚК әсерін электр өрісінің компоненті E , В/м немесе H , А/м магниттік компонентімен бағалайды		ЭҚК әсер етуі энергия ағынының тығыздығы P , Вт/м ² бойынша бағаланады

Жоғары температура және ультрафиолет тербелістерінің көздеріне қызмет көрсету үшін жұмыс орындары, әдетте, жақын аймақта орналасады, ал микротолқынды құрылғыларға қызмет көрсету жұмыстары алыс аймақта.

ГОСТ 12.1.006-84 сәйкес кернеудің рұқсат етілген ең жоғары деңгейлері анықталады: $E_{пд}$ немесе $H_{пд}$ немесе энергия ағынының тиімділігінің шекті деңгейінің рұқсат етілген деңгейі. Қауіп аймағының радиусы формулалардың біреуін пайдаланып есептеледі [8].

Егер есептелген нүкте (жұмыс орны) қауіпті аймақтан тыс жерде болса, онда қорғау шаралары қажет емес. Егер жұмыс орны қауіпті аймақта болса, онда сіз осы аймақта өткізілетін уақытты азайтыңыз немесе қорғаныс экранын орнатыңыз. Экранды есептеу процедурасы төменде келтірілген.

Электрмагниттік өрістің күші E немесе H есептеледі немесе формулалардың біріне сәйкес энергия ағынының тығыздығы есептеледі.

Қажетті экран тиімділігі анықталады:

$$\mathcal{E}_{mp} = \frac{E}{E_{пд}}, \quad (4.14)$$

$$\mathcal{E}_{mp} = \frac{H}{H_{пд}}, \quad (4.15)$$

немесе

$$\mathcal{E}_{mp} = \frac{P}{P_{пд}}, \quad (4.16)$$

мұнда \mathcal{E}_{tr} - ЭМӨ әсерін азайту үшін қанша рет қажет болатындығын көрсететін қажетті қорғаныс тиімділігі.

E , H , P - сәйкесінше өрістің электрлік және магниттік компоненттері және адамдар табуға болатын кеңістіктегі энергия ағынының тығыздығы;

$E_{пд}$, $H_{пд}$, $P_{пд}$ - өрістің электрлік және магниттік компоненттерінің шекті рұқсат етілген деңгейлері және энергия ағынының тығыздығы.

Экранның тиімділігі өте үлкен шектерде (бірліктен 10^{10} -ға дейін) өзгеруі мүмкін болғандықтан, тәжірибеде экрандау тиімділігі әдетте логарифмдік бірліктерде - децибельдерде (дБ) ұсынылады:

$$L_E = 20 \lg \frac{E}{E_{пд}}, \quad (4.17)$$

$$L_H = 20 \lg \frac{H}{H_{\text{пд}}}, \quad (4.18)$$

$$L_{\Pi} = 20 \lg \frac{\Pi}{\Pi_{\text{пд}}}. \quad (4.19)$$

Жобаланған Еекг экранының тиімділігі есептеледі. Есептеу ЭМӨ қамту аймағына ($r < 1 / 2p$ немесе алыс $r \gg 1 / 2p$ жақын) және экран түріне (қатты, тор және т.б.) байланысты формулаларға сәйкес жүзеге асырылады. Формулалар 4.21 бөлімінде келтірілген.

ЭМӨ-ні бағалау және реттеу энергия әсерінің мәні бойынша жүзеге асырылады. Электромагниттік өрістің энергияға әсер етуі электр немесе магнит өрісінің квадраты мен адамға әсер ету уақыты ретінде анықталады.

Формулаларды қолдана отырып, 30 кГц-300 МГц жиілік диапазонындағы энергияның әсерін есептеңіз:

$$\mathcal{E}_E = E^2 T, \quad (4.20)$$

$$\mathcal{E}_H = H^2 T, \quad (4.21)$$

мұндағы, E - электр өрісінің күші, В / м;

H - магнит өрісінің күші, А / м;

T - бір ауысымда жұмыс орнындағы әсер ету уақыты, сағ.

Егер экранның саңылаулары немесе қарау терезелері тормен жабылған, өткізгіш беті бар әйнек немесе бал торы түрінде жасалған болса, онда жалпы тексерудің тиімділігі осы қауіпті аймақпен анықталады. Шұңқырлардың, саңылаулардың, жұмыртқа торларының және т.б. максималды мөлшерін анықтауға арналған ұсыныстар. 2.7 бөлімінде берілген.

Формула бойынша 300 МГц - 300 ГГц жиілік диапазонындағы энергия ағынының тығыздығы бойынша энергия шығынын есептеңіз:

$$\mathcal{E}_H = H^2 T, \quad (4.22)$$

мұндағы \mathcal{E}_H - энергия ағынының тығыздығы (мкВт / см²).

Экранның қалыңдығы әдетте 0,8-ден 2 мм-ге дейінгі дәнекерлеу түрі мен беріктігі жағдайынан алынады. Жіңішке электр материалдарынан жасалған торлы экрандарды немесе экрандарды пайдаланған кезде, балама тордың қалыңдығы де немесе металл қабатының қалыңдығы электромагниттік өрістің ену тереңдігінің мәнінен үлкен болатындығын, яғни. $d > p, d$.

Жағдайы тексерілуі:

$$\mathfrak{E}_{\text{экp}}^3 \mathfrak{E}_{\text{mp}}, \quad (4.22)$$

экрандау құралы электромагниттік өрістің қолайлы мәнге дейін әлсіреуін қамтамасыз етеді.

Өлшенген электр өрісінің кернеулігі I және II диапазонында сәйкесінше 10-1000 В / м, 1 - 100 В / м.

Менің компьютерімнің алынатын электромагниттік кернеулігі – E=50 В/м. Стандарттарға сәйкес ЭМӨ 0,03 Гц - 3 МГц диапазонында өлшенеді.

СанПиН 2.1.8/2.24.1383-03 стандартынан өрістің электрлік және магниттік компоненттерінің шекті рұқсат етілген $E_{\text{ПД}}=500$ В/м.[7]

Энергияның ағынының тығыздығы $\mathfrak{E}_E = 20000$ В/м.[7]

$$\mathfrak{E}_{\text{mp}} = \frac{E}{E_{\text{ПД}}} = \frac{50}{500} = 0,1.$$

Қажетті экран тиімділігін (4.14) формуласы бойынша анықтаймыз.

$$T = \frac{\mathfrak{E}_E}{E^2} = \frac{20000}{2500} = 8 \text{ (сағ)}.$$

Бір ауысымда жұмыс орнындағы әсер ету уақыты, сағ (4.20) формуласы бойынша анықтаймыз.

4.3 Өміртіршілік қауіпсіздігі бөлімі бойынша қорытынды

Өміртіршілік қауіпсіздігі бөлімінде жұмыс орнында жұмысшыларға әсер ететін әртүрлі факторлар қарастырылды және осы факторлардың алдын алу мақсатында есептеулер жүргізілді. Сәйкесінше, Бір ауысымда жұмыс орнындағы әсер ету уақыты, 8 сағ. Әрі қарай, кондиционерлеу жүйесі есептелініп, арнайы кондиционер маркасы таңдалып алынды.

5 Техникалық-экономикалық бөлім

5.1 Жұмыстың мақсаты

Берілген дипломдық жобаның мақсаты телекоммуникация байланыс жүйесіндегі Дейксер әдісін зерттеу және талдау, сондай-ақ маршрутизаторлар арасындағы ең қысқа арақашықтықты есептеуге арналған бағдарлама жасау және оның көмегімен жұмысшыларға барынша жағдай жасау болып табылады. Маршрутизаторлардың пакетінің жүру жолын көрсету кезінде мына жасалынған бағдарлама тапшы болмақ.

5.2 Жұмыстың орындалу бағдарламасы

Байланыс жүйесін жобалау жұмысының өзіндік құнының шығын тармақтарына кіретін келесі бөлімдерді есептеу керек: негізгі іскерлердің еңбекақысы, қосымша еңбекақы, еңбекақы қоры, еңбекақы қорынан әлеуметтік салыққа аударма, амортизациялық аударма, оның ішінде коммуналды төлем бар, заттар шығыны, салықтар (жол, көлік т.б), үстеме шығындар. Осы жолмен жобалау жұмысы нәтижелерінің техникалық-экономикалық негізделуі болып табылады.

Бұл есептерді есептеу үшін берілген тақырыбыма сәйкес жұмыстың бизнес жоспарын құрастыру керек және оңтайландырылған желінің тиімділігін есептеу керек.

«Дейкстер алгоритмін зерттеп, оны программалық тілде моделдеу» тақырыбы бойынша бизнес жоспар құру үшін барлық есептеулерді кестеге енгізе отырып көрсетеміз.

5.3 Қаржылық жоспары

Қаржылық жоспарды – іске қосу үшін инвестициялық жобаны қабылдау үшін негізгі шарт ретінде бизнес жоспарды ең маңызды бөлімінің бірі болып саналады.

Қаржы жоспары жалпы күрделі шығындар, кірістер, операциялық шығындарды, пайданы және өзін-өзі ақтау кезеңінің есептеуін қарастырады.

Керек құрал-жабдықтардың тізімі 5.1-кестеде келтірілген. Ал 5.2-кестеде жобаға тартылған қызметкерлер туралы мәліметтер және олардың жалақысы туралы тізімі берілген.

5.1 кесте – Техникалық құрылғыларға кететін шығындар

Атауы	Сипаттамасы	Саны	Бір данасы үшін бағасы, теңге	Жалпы бағасы, теңге
Жүйелік блок	Ноутбук Acer Aspire E5 (Intel® Core™ i3 / 8GB/1TB/940MX 2GB/DVDRW)	1	165000	165000

5.1 кестенің жалғасы

Атауы	Сипаттам(асы	Саны	Бір данасы үшін теңге	Жалпы бағасы, теңге
Принтер	HP LaserJet MFP M125a	1	55000	55000
Бағдарламалық қамтамасыз ету	MS Windows10 Professional	1	60000	60000
Желдеткіш	Samsung AR12MSFPAWQNER White	1	160000	160000
Жалпы бағасы				437000

Жабдықтарға баға ҚҚС-сыз есептемей келтірілген.

5.2 кесте – Жобаға тартылған қызметкерлер туралы мәліметтер және олардың жалақысы

Атауы	Саны	Жалақысы, теңге
Әзірлеуші	1	80000
Жобаның менеджері	1	80000
Барлығы	2	160000

5.3.1 Жобаны іске асыру мерзімі

Бағдарламалық жасақтама өнімін жобалау мен әзірлеу кезеңдерден тұрады және мынадай жұмыс түрлерін қамтиды:

1-кезең – тапсырманы белгілеу, қажетті ақпаратты жинау, дипломдық жобаның құрылымын жасау;

2-кезең – жобаның бірінші бөлігін дамыту

3- кезең – математикалық қолдауды әзірлеу;

4-кезең – бағдарламалық қамтамасыз етуді әзірлеу;

5-кезең – имитациялық моделін тестілеу;

6-кезең – есептерді тіркеу.

Жоба әзірлеу кестесі 5.3-кестеде келтірілген.

5.3 кесте – Жобаны іске асыру кезеңдері мен мерзімі

Кезең атауы		Жұмыстың жасалауы апталарға бөлінген									
		1	2	3	4	5	6	7	8	9	10
1-этап	Тапсырма қоюлуы										
	Әдебиеттерді іріктеу және талдау										
	Жобаның құрылымын әзірлеу										

5.3 кестенің жалғасы

Кезең атауы		Жұмыстың жасалауы апталарға бөлінген									
		1	2	3	4	5	6	7	8	9	10
2этап	Жобаның бірінші бөлімін әзірлеу										
3этап	Математикалық қаматамасыз етуді әзірлеу										
4 этап	Есептеу бөлімін әзірлеу										
5этап	Презентацияны әзірлеу										
6этап	Тексеру және есеп беру										

5.3.2 Жалақы қорын есептеу

Жалақы қоры – барлық қызметкерлер үшін негізгі және қосымша жалақыдан қалыптасқан және белгілі бір мерзімге еңбек ақысын төлеуге арналған жалпы шығындар болып табылады және 5.1 формуласымен анықталады [9]:

$$EAK = ЖК_{н} + ЖК_{кoc}, \quad (5.1)$$

мұндағы $ЖК_{н}$ – негізгі жалақы;

$ЖК_{кoc}$ – қосымша жалақы.

БӨ-ді әзілеу кезеңінде жұмысшылар біркелкі емес түрде жұмыс істейді, бұл үшін орташа күндік жалақыны есептеу керек, содан кейін жалақының жалпы мөлшерін.

Әр қызметкердің орташа күндік жалақысын формула бойынша есептеледі:

$$D = ЖК_{a} / K_{жс}, \quad (5.2)$$

мұндағы $ЖК_{a}$ – айлық жалақы, теңге;

$K_{жс}$ – бір айдағы жұмыс күндерінің саны (22 күн – бес күндік жұмыс аптасы).

1.Әзірлеуші:

$$D = \frac{88000}{22} = 4000 (\text{теңге} / \text{кунге}).$$

2.Жоба менеджері:

$$D = \frac{88000}{22} = 4000 (\text{тенге} / \text{кунге}).$$

Бір сағатқа еңбекақысы формула бойынша есептеледі: күнге

$$H = JK_a / (K_{жс} \cdot C_{жс}), \quad (5.3)$$

мұндағы JK_a – айлық жалақы, теңге;

$K_{жс}$ – бір айдағы жұмыс күндерінің саны (22 күн – бес күндік жұмыс аптасы);

$C_{жс}$ – жұмыс күнінің ұзақтығы, сағат, $C_{жс} = 8$ сағат

1.Әзірлеуші:

$$D = \frac{88000}{22 \cdot 8} = 500 (\text{тенге} / \text{сағ}) \cdot$$

2.Жоба менеджері:

$$D = \frac{88000}{22 \cdot 8} = 500 (\text{тенге} / \text{сағ}).$$

Жұмыстың әрбір түріне арналған циклдің ұзақтығы формуламен анықталады:

$$t_n = \frac{T}{q_n \cdot z \cdot K} \quad (5.4)$$

мұндағы T – кезеңін еңбек қарқындылығы, норма-сағат;

q_n – орындаушылар саны;

z – жұмыс күнінің ұзақтығы, сағат $z = 8$;

K – уақыт нормаларына сәйкестік коэффициенті, $K = 1,1$.

Алынған t_n мән бүтін күндерге дейін жинақтаймыз:

Жетекші, тапсырма қою:

$$t_1 = \frac{20}{1 \cdot 8 \cdot 1.1} \approx 2 (\text{кун})$$

Жетекші, әзірлеуші, әдебиеттерді іріктеу және зерттеу:

$$t_2 = \frac{25 + 20}{2 \cdot 8 \cdot 1.1} \approx 3 (\text{кун})$$

Жетекші, әзірлеуші, жоба құрылымын әзірлеу:

$$t_3 = \frac{15 + 20}{3 \cdot 8 \cdot 1.1} \approx 2(\text{кун})$$

Әзірлеуші; техникалық бөлімді есептеу:

$$t_4 = \frac{88}{1 \cdot 8 \cdot 1.1} \approx 10(\text{кун})$$

Әзірлеуші, мәлімет жинау:

$$t_5 = \frac{44}{1 \cdot 8 \cdot 1.1} \approx 5(\text{кун})$$

Әзірлеуші, жүйені модельдеу:

$$t_6 = \frac{80}{1 \cdot 8 \cdot 1.1} \approx 9(\text{кун})$$

Әзірлеуші, жетекші, жоба презентациясын дайындау:

$$t_7 = \frac{20 + 24}{2 \cdot 8 \cdot 1.1} \approx 3(\text{кун})$$

Әзірлеуші, жетекші, тексеру және есеп беру:

$$t_8 = \frac{35}{2 \cdot 8 \cdot 1.1} \approx 2(\text{кун})$$

5.4 кесте – Негізгі жалақы бойынша шығындарды есептеудің жиынтық нәтижелері

Жұмыс атауы	Орындаушы	Еңбек қарқындылығы			Еңбек жалақысы, сағатына теңге	Жалақы мөлшері, теңге
		Норма -сағат	Жалпы еңбек сыйымдылығын ың%- ы	Цикл уақыты,күн		
Тапсырма қою	Жетекші	20	5.12	2	500	10000
Әдебиеттерді іріктеу және зерттеу	Жетекші	25	6.39	3	500	12500
	Әзірлеуші	20	5.12	2	500	10000

5.4 кестенің жалғасы

Жұмыс атауы	Орындаушы	Еңбек қарқындылығы			Еңбек жалақысы,	Жалақы мөлшері, теңге
		Норма -сағат	Жалпы еңбек сыйымды	Цикл уақыты, к		
Жобаның құрылымын әзілеу	Әзірлеуші	15	3.84	2	500	7500
	Жетекші	20	5.12	2	500	10000
Техникалық бөлімді есептеу	Әзірлеуші	88	22.51	11	500	44000
Мәлімет жинау	Әзірлеуші	44	11.25	5	500	22000
Жүйені модельдеу	Әзірлеуші	80	20.46	10	500	40000
Жоба презентациясын дайындау	Әзірлеуші	20	5.12	2	500	10000
	Жетекші	24	6.14	3	500	12000
Тексеру және есеп беру	Әзірлеуші	20	5.12	2	500	10000
	Жетекші	15	3,84	2	500	7500
Барлығы		391	100	46		195000

Қосымша жалақы негізгі жалақының 10%-ын құрайды және 5.5 формула бойынша есептеледі:

$$ЖК_{\text{кос}} = ЖК_{\text{нег}} \cdot 0,1 \quad (5.5)$$

$$ЖК_{\text{кос}} = 195000 \cdot 0,1 = 19500 \text{ (теңге)}$$

Осылайша, жалақының жалпы қоры болады:

$$ЕАК = 195000 + 19500 = 214500 \text{ (теңге)}$$

5.3.3 Әлеуметтік салық бойынша шығындарды есептеу

Әлеуметтік салық қызметкердің табысынан 9,5% (ҚР Салық кодексінің 358-бабы 1-тармағы) болып табылады және 5.6 формуласына сәйкес есептеледі [9]:

$$C_n = (ЕАК - 3Ж) \cdot 0,095 \quad (5.6)$$

Онда $ЗЖ$ – зейнетақы жарналары, 10% құрайды және әлеуметтік салық төленуге жатпайды.

$$ЗЖ = ЕАК \cdot 0,1 \quad (5.7)$$

$$ЗЖ = 214500 \cdot 0,1 = 21450 (\text{тенге})$$

Әлеуметтік мұқтаждықтарға аударымдардың сомасы:

$$O_{CH} = 9,5\% \cdot (ЕАК - ЗЖ) \quad (5.8)$$

$$C_H = (214500 - 21450) \cdot 0,095 = 18340 (\text{тенге})$$

Әлеуметтік аударымдар ЕАҚ-ның 9,5% құрайды:

$$Ш_{AA} = (ЕАК - ЗЖ) \cdot 9,5\% \quad (5.9)$$

$$Ш_{AA} = (214500 - 21450) \cdot 0,095 = 18340 (\text{теңге})$$

Міндетті әлеуметтік сақтандыруға арналған аударымдар (ЕАҚ-ның 1,5%):

$$Ш_{MACAA} = ЕАК \cdot 1,5\% \quad (5.10)$$

$$Ш_{MACAA} = 214500 \cdot 0,015 = 3217,5 \text{ теңге}$$

5.3.4 Амортизациялық аударымдарды есептеу

Амортизациялық аударымдар 5.11 формуласы бойынша есептеледі:

$$A_i = \frac{H_A \cdot C_{map} \cdot N}{100 \cdot 12 \cdot n} \quad (5.11)$$

мұндағы H_A – амортизация нормасы;

C_{map} – жабдықтың бастапқы құны;

N – жұмыс орындауына керек күндері саны;

n – жұмыс айындағы күндер саны.

Компьютерлік техника мен бағдарламалық қамсыздандыруға арналған H_A -ның құнсыздануының жалпы құнын 40% құрайды.

Жабдықтар мен пайдаланылған бағдарламалық жасақтаманың амортизациялық шығыны болады:

$$A_1 = (25 \cdot 165000 \cdot 46) / (100 \cdot 12 \cdot 22) = 7187 (\text{теңге})$$

$$\begin{aligned}
A_2 &= (25 \cdot 55000 \cdot 46) / (100 \cdot 12 \cdot 22) = 2396 (m_z) \\
A_3 &= (25 \cdot 60000 \cdot 46) / (100 \cdot 12 \cdot 22) = 2614 (m_z) \\
A_4 &= (25 \cdot 160000 \cdot 46) / (100 \cdot 12 \cdot 22) = 6970 (m_z) \\
A &= 7187 + 2396 + 2614 + 6970 = 19167 (m_z)
\end{aligned}$$

5.3.5 Электр энергиясының шығындарын есептеу

Электр жабдықтарын өндіру кезінде электр энергиясының құнын есептеу қажет. Өндірістік қажеттіліктерге арналған электр энергиясының құны жабдықтар мен қосымша қажеттілік үшін электр энергиясының құнын қамтиды.

$$\mathcal{E} = \mathcal{Z}_{\text{эл.эн.қур}} + \mathcal{Z}_{\text{қос.қаж}} \quad (5.12)$$

мұндағы $\mathcal{Z}_{\text{эл.эн.қур}}$ – жабдықтардың электр тұтыну құны;

$\mathcal{Z}_{\text{қос.қаж}}$ – қосымша қажеттілікке арналған энергия шығыны.

Жабдықтардың электр энергиясын тұтыну 5.14 формуласы бойынша есептеледі:

$$\mathcal{Z}_{\text{эл.эн.қур}} = W \cdot T \cdot S \cdot K_{\text{кол}} \quad (5.13)$$

мұндағы W – энергияны тұтыну, Вт;

T – жабдықтарды пайдалану сағаттарының саны;

S – киловатт-сағат электр энергиясының құны, (1 кВтсағ=24

теңге);

$K_{\text{кол}}$ – қолдану коэффициенті, ($K_{\text{кол}}=0,9$). $W = 90 \text{ Вт} = 0,09 \text{ кВт}$

(компьютердің қуаты);

$T = 46 \cdot 5 = 391$ сағат; $S = 24$ теңге.

Негізгі жабдықтарға арналған электр қуатының құны:

$$\mathcal{Z}_{\text{эл.энер}} = 0,09 \cdot 391 \cdot 24 \cdot 0,9 = 760 (\text{теңге})$$

Қосымша қажеттілікке арналған шығындар жабдықтың өзіндік құнын 5% мөлшерінде жиынтық индикатор бойынша қабылданады:

$$\mathcal{Z}_{\text{қос.қаж}} = 0,05 \cdot \mathcal{Z}_{\text{эл.эн.қур}} \quad (5.14)$$

$$\mathcal{Z}_{\text{қос.қаж}} = 0,05 \cdot 760 = 38 (m_z)$$

Электр энергиясының жалпы құны:

$$\mathcal{E} = 760 + 38 = 798 (m_z)$$

5.3.6 Үстеме шығындар

Жалпы шығындар 50% -ды құрайды және формула бойынша есептеледі:

$$ЖШ = (ЕАК + C_n + A + Э) \cdot 0,5 \quad (5.15)$$

5.15 формуласына сәйкес үстеме шығыстар:

$$ЖШ = (214500 + 18340 + 19167 + 798) \cdot 0.5 = 126402 (m_z)$$

Осылайша, 5.1 формуласына сәйкес, бағдарламалық жасақтама өнімін жасаудың жалпы құны:

$$Ж_{\kappa} = 214500 + 18340 + 19167 + 798 + 126402 = 379207 (m_z)$$

5.5 кесте – Жүйенің имитациялық моделін құру құны

Шығын атауы	Құны, теңге	Шығын мөлшері, %
Жалақы қоры	214500	56,8%
Әлеуметтік салық	18340	4,6%
Амортизациялық аударымдар	19167	5,1%
Электр қуаты шығындары	798	0,2%
Есептік шығындар	126402	33,3%
Жалпы құны	379207	100%

Имитациялық модельді құрылымын әзірлеуге жұмсалатын шығындар 5.6-кестеде ұсынылған.



5.1 сурет – Жүйенің имитациялық моделін құру құнының құрылымы

5.3.7 Жүзеге асыру құны

Бағдарламалық өнімді жүзеге асыру бағасы оның құндылығы мен пайдасынан тұрады:

$$Ц = Cб + П \quad (5.16)$$

мұндағы $Cб$ – өнімнің құны;
 $П$ – пайда.

Бастапқы бағаны анықтағанда, бағдарламалық өнімді іске асыру үшін кірістілік деңгейін (20%) белгілеу керек:

$$Ц_n = Cб \times (1 + P / 100) \quad (5.17)$$

мұндағы P – рентабельділік (20%).

$$Ц = 379207 \cdot (1 + 0.2) = 455050(тг)$$

Дайын өнімді іске асыру бағасы формула бойынша есептеледі:

$$Ц_p = Ц_n + НДС \quad (5.18)$$

Онда ҚҚС – қосылған құн салығы болып табылады.
 ҚҚС формула бойынша есептеледі:

$$НДС = Ц_p \cdot 0.12 \quad (5.19)$$

$$НДС = 455050 \cdot 0.12 = 54606(тг)$$

$$Ц_p = 455050 + 54600 = 509650(тг)$$

5.3.8 Экономикалық тиімділіктің көрсеткіштерін есептеу

5.6 кесте – Бағдарламасыз жұмыс жасаушы персоналдың айлық жалақысы:

Атауы	Саны	Жалақысы, (теңге)
Бағдарламалаушы	1	90000
Жобаның менеджері	1	85000
Инженер	1	75000
Барлығы	3	250000

5.7 кесте – Бағдарламаны қолданып жұмыс жасайтын персоналдың айлық жалақысы:

Атауы	Саны	Жалақысы, (теңге)
Әзірлеуші	1	88000
Жобаның менеджері	1	88000
Барлығы	2	176000

Талдауға және оңтайландыруға кететін соманың қайтарылу мерзімі келесі формуламен есептелінеді:

$$P_{\text{км}} = 12 \cdot (Ш_{\text{к}} / Э_{\text{жс}}) \quad (5.20)$$

мұндағы $Э_{\text{жс}}$ – жылдық экономия;

$P_{\text{км}}$ – қайтарылу мерзімі (аймен);

$Ш_{\text{к}}$ – бағдарламаны дайындауға кеткен шығындар;

$$P_{\text{км}} = (379207 / 888000) \cdot 12 = 5 \text{ ай}$$

5.8 кесте – Экономикалық үнемділік көрсеткіштерін есептеу

Шығындар	Бағдарламасыз (3 адам), (тенге)	Бағдарламамен (2 адам), (тенге)
Жылдық жалақы	3000000	2112000
Материалдық шығындар	437000	437000
Үстеме шығындар	126402	126402
Барлығы	3563402	2675402
Экономикалық үнемділігі	888000	

Жылдық үнемдеу және жылдық эксплуатациялық шығындарды біле отырып, АЖО-ны енгізумен тиімділік коэффициентін ($K_{\text{эф}}$) формула бойынша анықтауға болады:

$$K_{\text{эф}} = Э_{\text{жс}} / Ш_{\text{к}} \quad (5.21)$$

$$K_{\text{эф}} = Э_{\text{жс}} / Ш_{\text{к}} = 888000 / 379207 = 2.34$$

5.4 Экономика бөлімі бойынша қорытынды

Сонымен, жоба өзін 5 айда ақтап шығады. Бұл бағдарламаның артықшылығы пайдалану барысындағы қарапайымдылығы мен экономикалық тұрғыдан тиімділігі, сондай-ақ өнімділігі болып табылады. Жыл ішіндегі шығындар сомасы нақты өндірістік өзіндік құнды немесе жылдық эксплуатациялық шығындарды құрайды, менің жағдайымда эксплуатациялық шығындар 379207 теңгені құрады. Эксплуатациялық шығындар құрамындағы ең үлкен үлес – еңбекақы қоры 56,8%, сондай-ақ есептік шығындар – жалпы шығындардың 33,33% құрайды. Экономикалық тиімділікті есептеуді жасай отырып, біз бұл жобаның тиімділігі, күрделі салымдар мен ағымдағы шығындардың ең қолайлы комбинациясы беретін артықшылығы бар деп қорытынды жасауға болады.

Қорытынды

Бұл дипломдық жұмысты жасау барысында үйренген білімдерім біршама болды. Дипломдық жұмысымның мақсатына жеттім және де бұл жоба экономикалық тұрғыдан қарағанда пайда әкелді. Бұл жобаны жасау кезінде, кез-келген технологияны қолданудың өзіндік қауіпсіздік ережелерін қатаң сақтау керек екенін түсіндім. Сол себепті өміртіршілік қауіпсіздігі бөлімінде есептеулер жүргізіп, адам денсаулығына және табиғатқа зиян тигізбейтін методты ұсындым. Бағдарламаны қолдану әдістерін үйретіп, жалпы алгоритм қандай функцияларда қалай жұмыс жасайтынын программалық блог-сызбалар көмегімен түсіндірдім. Және де бағдарламаны жасау барысында қолданылған компоненттерге қысқаша сипаттама беріп кеттім. Бір мысалды теориялық және практикалық түрде графтарын сыздым. Екеуін салыстырдым. Салыстыру барысында ешқандай айырмашылық таппадым, сонда жасалынған бағдарламаны жүз пайызға дұрыс жасайтыннына анық көз жеткіздім.

Граф бойынша ең қысқа маршруттарды табуға арналған көптеген алгоритмдердің ішінен Дейкстр алгоритмінің сипаттамасын таптым. Бұл алгоритм графиктің барлық шыңдары мен олардың ұзындығы арасындағы ең қысқа жолдарды табады. Дейкстр алгоритмінің принципін сипаттадым, ол оңтайлы маршруттар мен олардың ұзындығын бір нақты шыңы және графиктің барлық басқа шыңдары арасындағы ұзындығын табады. Бұл алгоритмнің кемшілігі, егер графикте теріс салмақтың доғалары болса, ол дұрыс жұмыс істемейді. Осы жасылынған проект телекоммуникацияда жұмыс жасайтын жұмысшалардың өз істерін жасауды жеңілдетті. Программалық жасақтама жасылынды, мен оны барынша қолдануға ыңғайлы етіп жасадым. Жоба тәмамдалған соң қолданушылардың жеке пікірін алдым. Олардың жауабы қанағаттандыратын болды. Қолданушылардың бірі ЖШС «Wisk Telecom Solutions» . Мен одан әрі Қазақстандағы телекоммуникация саласындағы компьютерлік бағдарламаларды дамытуға барынша күш саламын.

Әдебиеттер тізімі

1 Жандаулетова, Ф. Р. Охрана труда : учебник для вузов / Ф.Р. Жандаулетова, Т.Е. Хакимжанов, Т.С. Санатова; МОН РК, НАО АУЭС. - Алматы : АУЭС, 2019. - 399 с.

2 СНиП РК 2.04-104-2012. Естественное и Искусственное освещение. Государственные нормативы в области архитектуры, градостроительства и строительства. – Астана, 2012.

3 МЕСТ 12.1.005-88 «Жұмыс аймағындағы ауа, жалпы санитарлық-гигиеналық талаптар» – Еңбекті қорғау стандарттарының жүйесі. Жұмыс аймағының ауасына жалпы санитарлық және гигиеналық талаптар. КСРО Денсаулық сақтау министрлігі, Бүкілодықтық кәсіподақтардың орталық кеңесімен 1988 жылы енгізілген.

4 Абдимуратов Ж.С., Мананбаева С. Безопасность жизнедеятельности. Методические указания к выполнению раздела “Расчет производственного освещения” в выпускных работах для всех специальностей бакалавриат. – Алматы: АИЭС, 2009.- 20с.

5 Абикенова А.А., Санатова Т.С. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Пожарная профилактика» в выпускных работах для всех специальностей. Бакалавриат - Алматы: АИЭС, 2009. - 32 с.

6 Т.Е.Хакимжанов, Ф.Р.Жандаулетова. БЖД. Дипломное проектирование. Расчет воздухопровода и выбор вентилятора. Методические указания по выполнению раздела в дипломных проектах для студентов всех форм обучения – Алматы: АИЭС, 2006. - 24 с.

7 Санитарные правила и нормы. СанПиН 2.1.8/2.24.1383-03. Электромагнитные излучения радиочастотного диапазона (ЭМИ РЧ). Госкомсанэпиднадзор России. – М.,2003.

8 Санатова Т.С., Хакимжанов Т.Е.. Охрана труда и основы безопасности жизнедеятельности. Защита от воздействия электромагнитных полей радиочастотного диапазона. Методические указания и задания к расчетно-графической работе для студентов специалистов 5В071900 - Алматы: АИЭС, 2010 - 33 с.

9 Бітіруші жұмысының экономикалық бөлімі үшін әдістемелік нұсқаулар. Радиодотехника, электроника және телекоммуникация мамандығының барлық оқу түрінің студенттеріне арналған. Алматы, 2009.

10 Зыков А. А. Основы теорий графов. М.: Наука, 1987г.

11 Евстигнеев В. А. Применение теорий графов в программировании / под редакцией Ершова А. П. –М: Наука, 1985 г

12 Салғараева Г. И. Графтар теориясы: Алматы: ЖШС «Дәуір», 2013. – 256-бет.