==========================================================

**Міністерство освіти і науки України**

**Національний технічний університет**

**«Дніпровська політехніка»**

****

**ЗВІТ**

**про виконання Курсової роботи**

**з дисципліни**

# «Дискретна Математика»

Виконав:

студент гр. 124-19-2

Моторний Андрій Сергійович

Прийняв:

Одновол Никлолай Николаевич

**Дніпро**

**2020**

**Вступ**

Дана Курсова Робота була створена для закріплення вивченого матеріалу в минулому році з предмету Дискретна Математика, вона складається з декількох Пунктів:

1. Розрахунок часових параметрів і визначення критичного шляху мережного графіка
   1. Розрахунок часових параметрів і визначення критичного шляху мережного графіка
   2. Розробка математичної моделі об’єкта дослідження
   3. Розрахунок параметрів математичної моделі
      1. Ранній термін настання події
      2. Пізній термін настання події
      3. Резерв часу по подіям
      4. Критичний шлях
      5. Ранній термін закінчення роботи
      6. Пізній термін закінчення роботи
      7. Повний резерв часу роботи
2. Оптимізація на мережах
   1. Алгоритм Дейкстра
   2. Алгоритм Форда – Фалкерсона
3. Дослідження методів мінімізації логічних функцій
   1. Мінімізіція логічної функції за допомогою карт Карно
   2. Аналітичний метод
   3. Порівняльний аналіз
4. Синтез кінцевого автомата по заданій таблиці переходів виходів
   1. Графічна структура кінцевого автомату
   2. Розробка загальної структури та комбінаційної схеми кінцевого автомату
5. Розробка програмного забезпечення для розрахунку параметрів математичної моделі оптимізації потоку в заданій мережі
   1. Розробка блок схеми
   2. Розробка програми
   3. Тестування програми
   4. Інструкція користувачу
6. Загальні висновки
7. Літературні джерела

**Розробка та дослідження математичної моделі дискретного об’єкта**

* 1. Опис об’єкта дослідження

Дуга

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1,2 | 1,4 | 2,7 | 2,5 | 2,3 | 4,3 | 4,6 | 4,9 | 3,5 | 3,6 | 5,7 | 5,8 | 6,7 | 6,8 | 7,10 | 8,10 | 8,9 | 9,10 |
| 4 | 1 | 6 | 11 | 7 | 6 | 2 | 4 | 5 | 6 | 3 | 5 | 7 | 4 | 9 | 2 | 9 | 11 |

Даний граф є Орієнтованим, пов'язаним.

1.2 Розробка математичної моделі об’єкта дослідження

**24**

**8**

**7**

**9**

**6**

**32**

**3**

**11**

**2**

**4**

**о**

**5**

**21**

**0**

**16**

**5**

**4**

**41**

**7**

**2**

**4**

**10**

**8**

**0**

**16**

**5**

**41**

**21**

**0**

**о**

**7**

**11**

**о**

**1**

**4**

**6**

**0**

**11**

**32естите здесь ваш текст**

**о**

**11**

**9**

**1**

**17**

**6**

**0**

**6**

**17**

**2**

**4**

**30**

**1**

**9**

**0**

**30**

**4**

**5**

**4**

1.3 Розрахунок параметрів математичної моделі

1) Ранній термін настання події i, tp (i) .Це максі-мінімальний шлях від початкового події до i - го події:

tp(1) = 0

tp(2) = tp(1) + t(1,2) = 0 + 4 = 4.  
tp(3) = tp(2) + t(2,3) = 4 + 7 = 11.  
tp(4) = tp(1) + t(1,4) = 0 + 1 = 1.  
tp(5) = max(tp(2) + t(2,5);tp(3) + t(3,5)) = max(4 + 11;11 + 5) = 16.  
tp(6) = max(tp(3) + t(3,6);tp(4) + t(4,6)) = max(11 + 6;1 + 2) = 17.  
tp(7) = max(tp(2) + t(2,7);tp(5) + t(5,7);tp(6) + t(6,7)) = max(4 + 6;16 + 3;17 + 7) = 24.  
tp(8) = max(tp(5) + t(5,8);tp(6) + t(6,8)) = max(16 + 5;17 + 4) = 21.  
tp(9) = max(tp(4) + t(4,9);tp(8) + t(8,9)) = max(1 + 4;21 + 9) = 30.  
tp(10) = max(tp(7) + t(7,10);tp(8) + t(8,10);tp(9) + t(9,10)) = max(24 + 9;21 + 2;30 + 11) = 41.

2) Пізній термін настання події i, tn (i) - це раз-ність між тривалістю максимального шляху lmax і шляхи найбільшої тривалості від даної події i до кінцевого події.

Розраховується tn (i) по зворотній схемі tp (i). Значить, розрахунок починаємо від кінцевого події, орієнтуємося на виходять роботи, беремо мінімум різниці.

tп(10) = tр(10) = 41

tп(9) = tп(10) - t(9,10) = 41 - 11 = 30

tп(8) = min(tп(9) - t(8,9);tп(10) - t(8,10)) = min(30 - 9;41 - 2) = 21

tп(7) = tп(10) - t(7,10) = 41 - 9 = 32

tп(6) = min(tп(7) - t(6,7);tп(8) - t(6,8)) = min(32 - 7;21 - 4) = 17

tп(5) = min(tп(7) - t(5,7);tп(8) - t(5,8)) = min(32 - 3;21 - 5) = 16

tп(3) = min(tп(6) - t(3,6);tп(5) - t(3,5)) = min(17 - 6;16 - 5) = 11

tп(4) = min(tп(3) - t(4,3);tп(6) - t(4,6);tп(9) – t(4,9)) = min(30 - 4;17 - 2;11 - 6) = 5

tп(2) = min(tп(3) - t(2,3);tп(5) - t(2,5); tп(7) - t(2,7)) = min(11 - 7;16 - 11;32 - 6) = 11

tп(1) = 0

3) Знаходимо резерв часу по подіям:

R( i ) = tn( i ) - tp( i ).

R(1) = 0; R(2) = 4 – 4 = 0; R(3) = 11 – 11 = 0; R(4) = 5 – 1 = 4; R(5) = 16 – 16 = 0;

R(6) = 17 – 17 = 0; R(7) = 32 – 24 = 8; R(8) = 21 – 21 = 0; R(9) = 30 – 30 = 0; R(10) = 41 – 41 = 0;

4)Критичний шлях проходить по подіях з нульовим резервом часу R (i) = 0, тобто 1, 2, 3, 5, 8, 9, 10. (виділено на графі). Довжина критичного шляху Lкр - це найдовший шлях від початкового події до кінцевого:

Lкр = tп (10) = 41

5) Ранній термін закінчення роботи (i, j):

tр.о(i , j) = tр(i) + t(i , j)

tр.о (i , j) = tр (i) + t(i , j)

tр.о (1, 2) = tр(1) + t(1 , 2) = 0 + 4 = 4

tр.о (1, 4) = tр (1) + t(1 , 4 )= 0 + 1 = 1

tр.о (2, 7) = tр (2) + t(2 , 7) = 4 + 6 = 10

tр.о (2, 5) = tр (2) + t(2 , 5) = 4 + 11 = 15

tр.о (2, 3) = tр (2) + t(2 , 3) = 4 + 7 = 11

tр.о (4, 3) = tр (4) + t(4 , 3) = 1 + 6 = 7

tр.о (4, 6) = tр (4) + t(4 , 6) = 1 + 2 = 3

tр.о (4, 9) = tр (4) + t(4 , 9) = 1 + 4 = 5

tр.о (3, 5) = tр (3) + t(3 , 5) = 11 + 5 = 16

tр.о (3, 6) = tр (3) + t(3 , 6) = 11 + 6 = 17

tр.о (5, 7) = tр (5) + t(5 , 7) = 16 + 3 = 19

tр.о (5, 8) = tр (5) + t(5 , 8) = 16 + 5 = 21

tр.о (6, 7) = tр (6) + t(6 , 7) = 17 + 7 = 24

tр.о (6, 8) = tр (6) + t(6 , 8) = 17 + 4 = 21

tр.о (7, 10) = tр (7) + t(7 , 10) = 24 + 9 = 33

tр.о (8, 10) = tр (8) + t(8 , 10) = 21 + 2 = 23

tр.о (8, 9) = tр (8) + t(8 , 9 )= 21 + 9 = 30

tр.о (9, 10) = tр (9) + t(9 , 10) = 30 + 11 = 41

6) Пізній термін закінчення роботи (i, j):

tп.о(i , j) = tп(i) + t(i , j)

tп.о (1, 2) = tп(1) + t(1 , 2) = 0 + 4 = 4

tп.о (1, 4) = tп(1) + t(1 , 4 )= 0 + 1 = 1

tп.о (2, 7) = tп(2) + t(2 , 7) = 4 + 6 = 10

tп.о (2, 5) = tп(2) + t(2 , 5) = 4 + 11 = 15

tп.о (2, 3) = tп(2) + t(2 , 3) = 4 + 7 = 11

tп.о (4, 3) = tп(4) + t(4 , 3) = 5 + 6 = 10

tп.о (4, 6) = tп(4) + t(4 , 6) = 5 + 2 = 7

tп.о (4, 9) = tп(4) + t(4 , 9) = 5 + 4 = 9

tп.о (3, 5) = tп(3) + t(3 , 5) = 11 + 5 = 16

tп.о (3, 6) = tп(3) + t(3 , 6) = 11 + 6 = 17

tп.о (5, 7) = tп(5) + t(5 , 7) = 16 + 3 = 19

tп.о (5, 8) = tп(5) + t(5 , 8) = 16 + 5 = 21

tп.о (6, 7) = tп(6) + t(6 , 7) = 17 + 7 = 24

tп.о (6, 8) = tп(6) + t(6 , 8) = 17 + 4 = 21

tп.о (7, 10) = tп(7) + t(7 , 10) = 32 + 9 = 41

tп.о (8, 10) = tп(8) + t(8 , 10) = 21 + 2 = 23

tп.о (8, 9) = tп(8) + t(8 , 9 )= 21 + 9 = 30

tп.о (9, 10) = tп(9) + t(9 , 10) = 30 + 11 = 41

7) Повний резерв часу роботи i, j - це час, на кото-рої можна збільшити тривалість даної роботи, не змінюючи при цьому тривалість критичного шляху Lкр.

Rn( i , j ) = tп ( j ) - tp ( i ) - t(i,j);

RП(1,2) = tп (2) - tp (1) - t(1,2) = 11 - 0 – 4 = 7;

RП(1,4) = tп (4) - tp (1) - t(1,4)= 5 - 0 – 1 = 4;  
RП(2,3) = tп (3) - tp (2) - t(2,3) = 11 - 4 – 7 = 0;  
RП(2,5) = tп (5) - tp (2) - t(2,5) = 16- 4 – 11 = 1;  
RП(2,7) = tп (7) - tp (2) - t(2,7) = 32- 4 – 6 = 22;  
RП(3,5) = tп (5) - tp (3) - t(3,5) = 11- 11 – 5 = -5;  
RП(3,6) = tп (6) - tp (3) - t(3,6) = 17- 11 – 6 = 0;  
RП(4,3) = tп (4) - tp (3) - t(4,3) = 5- 11 – 6 = -12;  
RП(4,6) = tп (6) - tp (4) - t(4,6) = 17- 1 – 2 = 14;  
RП(4,9) = tп (9) - tp (4) - t(4,9) = 30- 1 – 4 = 25;  
RП(5,7) = tп (7) - tp (5) - t(5,7) = 32- 16 – 3 = 13;  
RП(5,8) = tп (8) - tp (5) - t(5,8) = 21- 16 – 5 = 0;  
RП(6,7) = tп (7) - tp (6) - t(6,7) = 32- 17 – 7 = 8;  
RП(6,8) = tп (8) - tp (6) - t(6,8) = 21- 17 – 4 = 0;  
RП(7,10) = tп (10) - tp (7) - t(7,10) = 41- 24 – 9 = 8;  
RП(8,9) = tп (9) - tp (8) - t(8,9) = 30- 21 – 9 = 0;  
RП(8,10) = tп (10) - tp (8) - t(8,10) = 41- 21 – 2 = 18;  
RП(9,10) = tп (10) - tp (9) - t(9,10) = 41- 30 – 11 = 0;

**Оптимізація на мережах**

2.1 Алгоритм Дейкстра

**F**

**9**

**6**

**3**

**a**

**11**

**Cток**

**5**

**g**

**d**

**7**

**2**

**4**

**t**

**5**

**s**

**7**

**b**

**4**

**6**

**ИCток**

**11**

**9**

**1**

**e**

**6**

**2**

**4**

**h**

**c**

1. Забарвимо вершину s.

Покладемо d(s) = 0;

d(a) = d(b) = d(c) = d(e) =d(f) = d(g) = d(h) =∞;

2. Поточна змінна y=s

d(a) = min { d(a), d(s) + d(s,a)} = min {∞; 0 + 4} = 4;

d(b) = min {d(b), d(s) + d(s,b)} = min {∞; 0 + ∞} = ∞;

d(c) = min {d(c), d(s) + d(s,c)} = min {∞; 0 + 1 } = 1;

d(d) = min {d(d), d(s) + d(s,d)} = min {∞; 0 + ∞ } = ∞;

d(e) = min {d(e), d(s) + d(s,e)} = min {∞; 0 + ∞ } = ∞;

d(f) = min {d(f), d(s) + d(s,f)} = min {∞; 0 + ∞ } = ∞;

d(g) = min {d(g), d(s) + d(s,g)} = min {∞; 0 + ∞ } = ∞;

d(h) = min {d(h), d(s) + d(s,h)} = min {∞; 0 + ∞ } = ∞;

d(t) = min {d(t), d(s) + d(s,t)} = min {∞; 0 + ∞ } = ∞;

min {d(a), d(b), d(c), d(d), d(e), d(f), d(g), d(h),d(t)} =

= min {4; ∞; 1; ∞; ∞; ∞; ∞; ∞; ∞} = 6.2; d(c) = 1 ;

3. Поточна змінна y=с

**s**

**c**

d(a) = min { d(a), d(c) + d(c,a)} = min {4; 1 + ∞} = 4;

d(b) = min {d(b), d(c) + d(c,b)} = min {∞; 1 + 6} = 7;

d(d) = min {d(d), d(c) + d(c,d)} = min {∞; 1 + ∞ } = ∞;

d(e) = min {d(e), d(c) + d(c,e)} = min {∞; 1 + 2 } = 3;

d(f) = min {d(f), d(c) + d(c,f)} = min {∞; 1 + ∞ } = ∞;

d(g) = min {d(g), d(c) + d(c,g)} = min {∞; 1 + ∞ } = ∞;

d(h) = min {d(h), d(c) + d(c,h)} = min {∞; 1 + 4 } = 5;

d(t) = min {d(t), d(c) + d(c,t)} = min {∞; 1 + ∞ } = ∞;

min {d(a), d(b), d(d), d(e), d(f), d(g), d(h),d(t)} =

= min {4; 7; ∞; 3; ∞; ∞; 5; ∞} = 6.2; d(e) = 3 ;

**s**

**c**

**e**

4. Поточна змінна y=e

d(a) = min { d(a), d(e) + d(e,a)} = min {4; 3 + ∞} = 4;

d(b) = min {d(b), d(e) + d(e,b)} = min {7; 3 + ∞} = 7;

d(d) = min {d(d), d(e) + d(e,d)} = min {∞; 3 + ∞ } = ∞;

d(f) = min {d(f), d(e) + d(e,f)} = min {∞; 3 + 7 } = 10;

d(g) = min {d(g), d(e) + d(e,g)} = min {∞; 3 + 4 } = 7;

d(h) = min {d(h), d(e) + d(e,h)} = min {5; 3 + ∞ } = 5;

d(t) = min {d(t), d(e) + d(e,t)} = min {∞; 3 + ∞ } = ∞;

min {d(a), d(b), d(e), d(f), d(g), d(h),d(t)} =

= min {4; 7; ∞; 10; 7; 5; ∞} = 4; d(a) = 4 ;

**s**

**c**

**e**

**A**

5. Поточна змінна y=a

d(b) = min {d(b), d(a) + d(a,b)} = min {7; 4 + 7} = 7;

d(d) = min {d(d), d(a) + d(a,d)} = min {∞; 4 + 11 } = 15;

d(f) = min {d(f), d(a) + d(a,f)} = min {10; 4 + 6 } = 10;

d(g) = min {d(g), d(a) + d(a,g)} = min {7; 4 + ∞ } = 7;

d(h) = min {d(h), d(a) + d(a,h)} = min {5; 4 + ∞ } = 5;

d(t) = min {d(t), d(a) + d(a,t)} = min {∞; 4 + ∞ } = ∞;

min {d(a), d(b), d(e), d(f), d(g), d(h),d(t)} =

= min {7; 15; 10; 7; 5; ∞} = 5; d(h) = 5 ;

**s**

**c**

**e**

**A**

**h**

6. Поточна змінна y=h

d(b) = min {d(b), d(h) + d(h,b)} = min {7; 5 + ∞ } = 7;

d(d) = min {d(d), d(h) + d(h,d)} = min {15; 5 + ∞ } = 15;

d(f) = min {d(f), d(h) + d(h,f)} = min {10; 5 + ∞ } = 10;

d(g) = min {d(g), d(h) + d(h,g)} = min {7; 5 + ∞ } = 7;

d(t) = min {d(t), d(h) + d(h,t)} = min {∞; 5 + 11 } = 16;

min {d(a), d(b), d(e), d(f), d(g), d(t)} =

= min {7; 15; 10; 7; 16} = 7; d(b) = 7 ; d(g) = 7

**s**

**c**

**e**

**A**

**h**

**b**

7. Поточна змінна y=b

d(d) = min {d(d), d(b) + d(b,d)} = min {15; 7 + 5 } = 12;

d(f) = min {d(f), d(b) + d(b,f)} = min {10; 7 + ∞ } = 10;

d(g) = min {d(g), d(b) + d(b,g)} = min {7; 7 + ∞ } = 7;

d(t) = min {d(t), d(b) + d(b,t)} = min {16; 7 + ∞ } = 16;

min {d(a), d(b), d(e), d(f), d(g), d(t)} =

= min {12; 10; 7; 16} = 5; d(g) = 7 ;

**s**

**c**

**e**

**A**

**h**

**b**

**g**

8. Поточна змінна y=g

d(d) = min {d(d), d(a) + d(a,d)} = min {12; 7 + } = 12;

d(f) = min {d(f), d(a) + d(a,f)} = min {10; 7 + ∞ } = 10;

d(t) = min {d(t), d(a) + d(a,t)} = min {16; 7 + 2 } = 16;

min {d(a), d(b), d(e), d(f), d(g), d(t)} =

= min {12; 10;; 16} = 5; d(t) = 9;

**2**

**4**

**2**

**1**

**s**

**c**

**e**

**A**

**h**

**b**

**g**

**t**

Висновок: Найкоротший шлях з витоку s в стік t тільки один,

складається з дуг (s, c) , (c,e) , (e,g) і (g, t) і дорівнює 9 одиниць.

* 1. Алгоритм Форда – Фалкерсона

**i**

**F**

**9**

**6**

**i**

**i**

**3**

**i**

**a**

**11**

**i**

**5**

**i**

**i**

**i**

**g**

**d**

**7**

**2**

**4**

**i**

**t**

**5**

**i**

**s**

**7**

**i**

**6**

**b**

**4**

**i**

**11**

**9**

**1**

**i**

**i**

**e**

**i**

**i**

**6**

**i**

**2**

**i**

**4**

**h**

**c**

Обозначено: I - ресурсы не використані

R - ресурсы використані повністю

IR - ресурсы використані частково

1)Маршрут (s,c),(c,h),(h,t)

P1=min{f(s,c),f(c,h),f(h,t)}=min{1,4,11}=1

**i**

**F**

**9**

**6**

**i**

**i**

**3**

**i**

**a**

**11**

**i**

**5**

**i**

**i**

**i**

**g**

**d**

**7**

**2**

**4**

**i**

**t**

**5**

**i**

**s**

**7**

**i**

**6**

**b**

**4**

**i**

**10**

**9**

**i**

**R**

**e**

**iR**

**i**

**6**

**i**

**2**

**iR**

**3**

**h**

**c**

2)Маршрут (s,a),(a,b),(b,e),(e,g),(g,h),(h,t)

P2=min{f(s,a),f(a,b),,f(b,e),f(e,g),f(g,h),f(h,t)}=min{4,7,6,4,9,11}=4

PΣ=5

**i**

**F**

**9**

**6**

**i**

**i**

**3**

**i**

**a**

**11**

**R**

**i**

**5**

**i**

**i**

**g**

**d**

**3**

**2**

**ir**

**t**

**5**

**s**

**7**

**i**

**2**

**b**

**R**

**ir**

**6**

**5**

**i**

**R**

**e**

**ir**

**iR**

**6**

**i**

**2**

**iR**

**3**

**h**

**c**

Висновок : Максимальний потік буде дорівнює 5 одиниць

Дослідження методів мінімізації логічних функцій

3.1 Мінімізіція логічної функції за допомогою карт Карно

Таблица Истинности

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x1 | x2 | x3 | x1x2 | ⌐x3 | x1x2 ⌐x3 | ⌐x2 | x1⌐x2 | x1⌐x2x3 | x1x2⌐x3 V x1⌐x2x3 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| x1x2⌐x3 V x1⌐x2x3 V x3 | x1x2⌐x3 V x1⌐x2x3 V x3 V ⌐x2 | ⌐x1 | ⌐x1⌐x2 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 |

|  |
| --- |
| x1x2⌐x3 V x1⌐x2x3 V x3 V ⌐x2 V ⌐x1⌐ x2 |
| 1 |
| 1 |
| 0 |
| 1 |
| 1 |
| 1 |
| 1 |
| 1 |

Карта Карно

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x1\x2x3 | 00 | 01 | 11 | 10 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x1\x2x3 | 00 | 01 | 11 | 10 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

K1 = ⌐x2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x1\x2x3 | 00 | 01 | 11 | 10 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

K2 = x3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x1\x2x3 | 00 | 01 | 11 | 10 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

K3 = x1

Об'єднаємо їх за допомогою операції АБО і отримаємо мінімізовану ДНФ:

x1⌐x2 x3

3.2 Аналітичний метод(Квайна)

3.2.1 Операція попарного неповного склеювання.

Порівнюємо попарно всі кон'юнкції (минтермов 3-го рангу)

|  |  |  |
| --- | --- | --- |
| № | Склеювання | Результат |
| 0 | {0,1} | ⌐x1⌐x2 |
| 1 | {0,4} | ⌐x2⌐x3 |
| 2 | {1,3} | ⌐x1x3 |
| 3 | {1,5} | ⌐x2x3 |
| 4 | {3,7} | x2x3 |
| 5 | {4,5} | x1⌐x2 |
| 6 | {4,6} | x1⌐x3 |
| 7 | {5,7} | x1x3 |
| 8 | {6,7} | x1x2 |

Порівнюємо попарно всі кон'юнкції (минтермов 2-го рангу)

|  |  |  |
| --- | --- | --- |
| № | Склеювання | Результат |
| 0 | {0,5} | ⌐x2 |
| 1 | {1,3} | ⌐x2 |
| 2 | {2,7} | x3 |
| 3 | {3,4} | x3 |
| 4 | {5,8} | x1 |
| 5 | {6,7} | x1 |

Аналізуючи отриману формулу, можна помітити, що минтермов у формулі повторюються і, відповідно до закону повторення, повторювані члени можуть бути видалені.

Етап II. Операція поглинання (покриття).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ⌐x1⌐x2⌐x3 | ⌐x1⌐x2x3 | ⌐x1x2x3 | x1⌐x2⌐x3 | x1⌐x2x3 | x1x2⌐x3 | x1x2x3 |
| ⌐x2 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| x3 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| x1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

Ядро : F = ⌐x2 V x3 V x1

* 1. Порівняльний аналіз

Метод Карт Карно більш громіздкий, ніж Аналітичний, але він може застосовуватися для 1-5 змінних, в той час як Аналітичний метод - тільки для 1-3 змінних, інакше методи будуть працювати неточно

**Синтез кінцевого автомата по заданій таблиці переходів виходів**

Графічна структура кінцевого автомату

Розробка загальної структури та комбінаційної схеми кінцевого автомату

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вхід\Стан | 0 | 1 | 2 | 3 |
| 0 | 2/2 | 2/1 | 1/3 | 1/0 |
| 1 | 2/0 | 2/1 | 3/1 | 2/0 |
| 2 | 3/2 | 2/3 | 1/0 | 2/1 |
| 3 | 1/1 | 0/3 | 2/2 | 3/0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X(n) | 0000 | 1111 | 2222 | 3333 |
| S(n) | 0123 | 0123 | 0123 | 0123 |
| S(n+1) | 2231 | 2220 | 1312 | 1223 |
| Y(n) | 2021 | 1133 | 3102 | 0010 |

1

2/3 v 3/0

0

0/0 v 1/1 v 3/0

0/2 v 1/1

0/1

2/1

1/3

2/0

2/2

3

2

0/2 v 3/0

3/0

1/3 v 3/1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| X1(n) | X2(n) | S1(n) | S2(n) | S(n+1) | | Y(n) |
| S1(n+1) | S2(n+1) |
| 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 2 |
| 0 | 0 | x | x | x | x | x |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | x | x | x | x | x |
| 0 | 0 | 1 | 0 | 1 | 0 | 3 |
| 0 | 0 | 1 | 1 | 0 | 0 | 3 |
| 0 | 0 | 0 | 0 | 0 | 1 | 3 |
| 1 | 0 | x | x | x | x | x |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 2 |
| 1 | 1 | x | x | x | x | x |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | x | x | x | x | x |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |

S1(n+1)= ⌐x1⌐x2⌐s1⌐s2 v ⌐x1⌐x2⌐s1s2 v ⌐x1⌐x2s1⌐s2 v ⌐x1x2⌐s1⌐s2 v ⌐x1x2⌐s1s2 v ⌐x1⌐x2s1⌐s2 v x1x2⌐s1s2 v x1x2s1s2 v x1⌐x2⌐s1s2 v x1⌐x2s1⌐s2 v v x1x2s1s2

S2(n+1)= ⌐x1⌐x2s1⌐s2 v ⌐x1x2s1s2 v ⌐x1⌐x2⌐s1⌐s2 v x1x2⌐s1s2 v v x1x2s1⌐s2 v x1⌐x2⌐s1⌐s2 v x1x2s1s2

Y(n)= ⌐x1⌐x2⌐s1⌐s2 v ⌐x1⌐x2⌐s1⌐s2 v ⌐x1⌐x2s1⌐s2 v ⌐x1⌐x2s1⌐s2 v

v ⌐x1x2s1s2 v ⌐x1x2⌐s1⌐s2 v ⌐x1x2⌐s1s2 v ⌐x1⌐x2⌐s1⌐s2 v ⌐x1⌐x2s1⌐s2 v

v ⌐x1⌐x2s1⌐s2 v ⌐x1⌐x2s1⌐s2 v ⌐x1⌐x2s1s2 v ⌐x1⌐x2s1s2 v ⌐x1⌐x2s1s2 v

v ⌐x1⌐x2⌐s1⌐s2 v ⌐x1⌐x2⌐s1⌐s2 v ⌐x1⌐x2⌐s1⌐s2 v x1x2⌐s1s2 v

v ⌐x1⌐x2⌐s1⌐s2 v x1x2s1s2 v x1x2s1s2 v ⌐x1x2⌐s1s2

Для S1(n+1)

X1

X2

|  |  |  |  |
| --- | --- | --- | --- |
| 1  S1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 |
| 1  S2 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |

F=⌐S1S2 v X1X2 v ⌐X1⌐X2⌐S2 v ⌐X2S1⌐S2 v ⌐X1⌐S1

Для S2(n+1)

X1

X2

|  |  |  |  |
| --- | --- | --- | --- |
| 0  S1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0  S2 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 |

F=⌐X1⌐S1⌐S2 v X1X2S2 v ⌐X1⌐X2⌐S2 v X1X2S1 v X2S1S2

Для Y(n)

X1

X2

|  |  |  |  |
| --- | --- | --- | --- |
| 0  S1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |
| 1  S2 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 |

F=⌐X1⌐S1

X1(n)

Комб. Схема

X2(n)

П2

П1

⌐X1

⌐X2

⌐S1

⌐S2

S1(n)

S2(n)

S2(n+1)

X1

S2

S1

X2

&

X1

X2

1

Y(n)

1

1

1

&

&

1

&

S2(n+1)

&

1

&

&

&

S1(n+1)

&

1

&

&

П1

S1

П2

S1

**Розробка програмного забезпечення для розрахунку параметрів математичної моделі оптимізації потоку в заданій мережі**

4.1 Розробка блок схеми

**Початок**

Int f[MAX\_VERTICES][MAX\_VERTICES], c[MAX\_VERTICES][MAX\_VERTICES], Flow[MAX\_VERTICES], Link[MAX\_VERTICES], Queue[MAX\_VERTICES], QP, QC

int FindPath(int source, int target);QP = 0; QC = 1; Queue[0] = source;Link[target] = -1; int i; int CurVertex;memset(Flow, 0, sizeof(int) \* NUM\_VERTICES);Flow[source] = infinity;

-

Link[target] == -1 && QP < QC

+

-

c[CurVertex][i] - f[CurVertex][i]) > 0 && Flow[i] == 0

+

Queue[QC] = i; QC++;

Link[i] = CurVertex;

-

+

c[CurVertex][i] - f[CurVertex][i] < Flow[CurVertex]

Flow[i] = c[CurVertex][i];

Flow[i] = Flow[CurVertex]

CurVertex != source

f[Link[CurVertex]][CurVertex] += Flow[target];

CurVertex=Link[CurVertex];

Кінець

printf("%d", MaxFlow(source, target))

int source, target,a,b;

cin<<a,b;

source = a; target = b

**Початок**

Введення ребер і їх ваг

Алгоритм Форда-Фалкерсона

Виведення максимального потоку

Кінець

**Код програми**

#include <iostream>

#include <memory.h>

#include <stdio.h>

#include <conio.h>

#include <Windows.h>

using namespace std;

const int MAX\_VERTICES = 40;

int NUM\_VERTICES;

const int infinity = 10000;

int f[MAX\_VERTICES][MAX\_VERTICES];

int c[MAX\_VERTICES][MAX\_VERTICES];

int Flow[MAX\_VERTICES];

int Link[MAX\_VERTICES];

int Queue[MAX\_VERTICES];

int QP, QC;

int FindPath(int source, int target)

{

QP = 0; QC = 1; Queue[0] = source;

Link[target] = -1;

int i;

int CurVertex;

memset(Flow, 0, sizeof(int) \* NUM\_VERTICES);

Flow[source] = infinity;

while (Link[target] == -1 && QP < QC)

{

CurVertex = Queue[QP];

for (i = 0; i < NUM\_VERTICES; i++)

if ((c[CurVertex][i] - f[CurVertex][i]) > 0 && Flow[i] == 0)

{

Queue[QC] = i; QC++;

Link[i] = CurVertex;

if (c[CurVertex][i] - f[CurVertex][i] < Flow[CurVertex])

Flow[i] = c[CurVertex][i];

else

Flow[i] = Flow[CurVertex];

}

QP++;

}

if (Link[target] == -1)

return 0;

CurVertex = target;

while (CurVertex != source)

{

f[Link[CurVertex]][CurVertex] += Flow[target];

CurVertex = Link[CurVertex];

}

return Flow[target];

}

int MaxFlow(int source, int target)

{

memset(f, 0, sizeof(int) \* MAX\_VERTICES \* MAX\_VERTICES);

int MaxFlow = 0;

int AddFlow;

do

{

AddFlow = FindPath(source, target);

MaxFlow += AddFlow;

} while (AddFlow > 0);

return MaxFlow;

}

int main()

{

setlocale(0, "");

int source, target;

printf("Число вершин в графе \n--> ");

NUM\_VERTICES = 11; printf("%d", NUM\_VERTICES-1);

cout << endl;

int a, b;

cout << "Введите Значение истока ";

cin >> a;

printf("\n Значение истока \n--> ");

source = a; printf("%d", source);

cout << endl;

cout << "Введите Значение стока ";

cin >> b;

printf("\n Значение стока \n--> ");

target = b; printf("%d", target);

cout << endl;

c[0][0] = 0; c[0][1] = 0; c[0][2] = 0; c[0][3] = 0; c[0][4] = 0; c[0][5] = 0; c[0][6] = 0; c[0][7] = 0; c[0][8] = 0; c[0][9] = 0; c[0][10] = 0;

c[1][0] = 0; c[1][1] = 0; c[1][2] = 4; c[1][3] = 0; c[1][4] = 1; c[1][5] = 0; c[1][6] = 0; c[1][7] = 0; c[1][8] = 0; c[1][9] = 0; c[1][10] = 0;

c[2][0] = 0; c[2][1] = 0; c[2][2] = 0; c[2][3] = 7; c[2][4] = 0; c[2][5] = 11; c[2][6] = 0; c[2][7] = 6; c[2][8] = 0; c[2][9] = 0; c[2][10] = 0;

c[3][0] = 0; c[3][1] = 0; c[3][2] = 0; c[3][3] = 0; c[3][4] = 0; c[3][5] = 5; c[3][6] = 6; c[3][7] = 0; c[3][8] = 0; c[3][9] = 0; c[3][10] = 0;

c[4][0] = 0; c[4][1] = 0; c[4][2] = 0; c[4][3] = 6; c[4][4] = 0; c[4][5] = 0; c[4][6] = 2; c[4][7] = 0; c[4][8] = 0; c[4][9] = 4; c[4][10] = 0;

c[5][0] = 0; c[5][1] = 0; c[5][2] = 0; c[5][3] = 0; c[5][4] = 0; c[5][5] = 0; c[5][6] = 0; c[5][7] = 3; c[5][8] = 5; c[5][9] = 0; c[5][10] = 0;

c[6][0] = 0; c[6][1] = 0; c[6][2] = 0; c[6][3] = 0; c[6][4] = 0; c[6][5] = 0; c[6][6] = 0; c[6][7] = 7; c[6][8] = 4; c[6][9] = 0; c[6][10] = 0;

c[7][0] = 0; c[7][1] = 0; c[7][2] = 0; c[7][3] = 0; c[7][4] = 0; c[7][5] = 0; c[7][6] = 0; c[7][7] = 0; c[7][8] = 0; c[7][9] = 0; c[7][10] = 9;

c[8][0] = 0; c[8][1] = 0; c[8][2] = 0; c[8][3] = 0; c[8][4] = 0; c[8][5] = 0; c[8][6] = 0; c[8][7] = 0; c[8][8] = 0; c[8][9] = 9; c[8][10] = 2;

c[9][0] = 0; c[9][1] = 0; c[9][2] = 0; c[9][3] = 0; c[9][4] = 0; c[9][5] = 0; c[9][6] = 0; c[9][7] = 0; c[9][8] = 0; c[9][9] = 0; c[9][10] = 11;

c[10][0] = 0; c[10][1] = 0; c[10][2] = 0; c[10][3] = 0; c[10][4] = 0; c[10][5] = 0; c[10][6] = 0; c[10][7] = 0; c[10][8] = 0; c[10][9] = 0; c[10][10] = 0;

printf("\n Максимальный поток равен: ");

printf("%d", MaxFlow(source, target));

\_getch();

return 0;

}

4.3 Тестування програми



4.4 Інструкція користувачу

Розпакувати файл Project14.rar --> Зайти у папку Project 14 --> потім у папку Debug --> Відкрити файл "Project 14.exe" --> Ввести значення Стоку (від 1-го до 10) --> Ввести значення Витоку (від 1-го до 10) --> Радіти отриманим даним

**Висновок**

Алгоритм або метод Форда-Фалкерсона знаходить максимальний потік у транспортній мережі.

Метод Форда-Фалкерсона - метод, який базується на трьох концепціях: залишкові мережі, шляхи що збільшуються і розрізи.[1] Ключову роль у методі Форда-Фалкерсона грають два поняття: залишкові мережі і доповнюють шляху.  Дані концепції лежать в основі важливої теореми про максимальний потік[en] і мінімальний розріз, яка визначає значення максимального нащадка за допомогою розрізів траспортної мережі. Метод Форда-Фалкерсона є ітеративним.

Спочатку величині потоку присвоюється значення 0: {\displaystyle f(u,v)=0} при будь-яких {\displaystyle u}, {\displaystyle v} Є {\displaystyle V}. На кожній ітерації величина потоку збільшується за допомогою пошуку «шляху, що збільшується» (тобто деякого шляху від джерела {\displaystyle s} до стоку {\displaystyle t}, уздовж якого можна послати більший потік) і подальшого збільшення потоку. Цей процес повторюється до тих пір, поки вже неможливо буде відшукати збільшуючийся шлях.

**Загальний Висновок**

Ця курсова робота дозволила нам згадати весь матеріал, з яким ми навчалися в минулому році по предмету Дискретна Математика, розширила наші знання в програмуванні алгоритмів і навчила нас правильно підносити

зроблену роботу та вмінню захищати її   
перед аудиторією.

**Джерела**

1)<https://studbooks.net/2222914/informatika/algoritm_forda_falkersona>

2)<https://graphonline.ru/?graph=SLHHzbczMQEuHebW>

3)<https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A4%D0%BE%D1%80%D0%B4%D0%B0_%E2%80%94_%D0%A4%D0%B0%D0%BB%D0%BA%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0>

4)<https://perevod.i.ua/ukrainsko-russkiy/>

5)<https://drive.google.com/file/d/13zdxrnaoCCAPNUPC1Xqh-3mwjOGUZBye/view?usp=sharing>

6)<https://www.cyberforum.ru/>

7)<https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%80%D1%82%D0%B0_%D0%9A%D0%B0%D1%80%D0%BD%D0%BE>

8)<https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%94%D0%B5%D0%B9%D0%BA%D1%81%D1%82%D1%80%D1%8B>

9) <http://csd.faculty.ifmo.ru/files/karnaugh.pdf>

10) <https://vunivere.ru/work4703/page7>

11)<https://ru.bmstu.wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BD%D1%8B%D0%B9_%D1%81%D0%B8%D0%BD%D1%82%D0%B5%D0%B7_%D0%BA%D0%BE%D0%BD%D0%B5%D1%87%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B0>

12) http://www.bibl.nngasu.ru/electronicresources/uch-metod/mathematics/4973.pdf