

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту

Розрахункова робота

з дисципліни

«Дискретна математика»

Виконала:

студентка групи КН-112

Максимець Віра

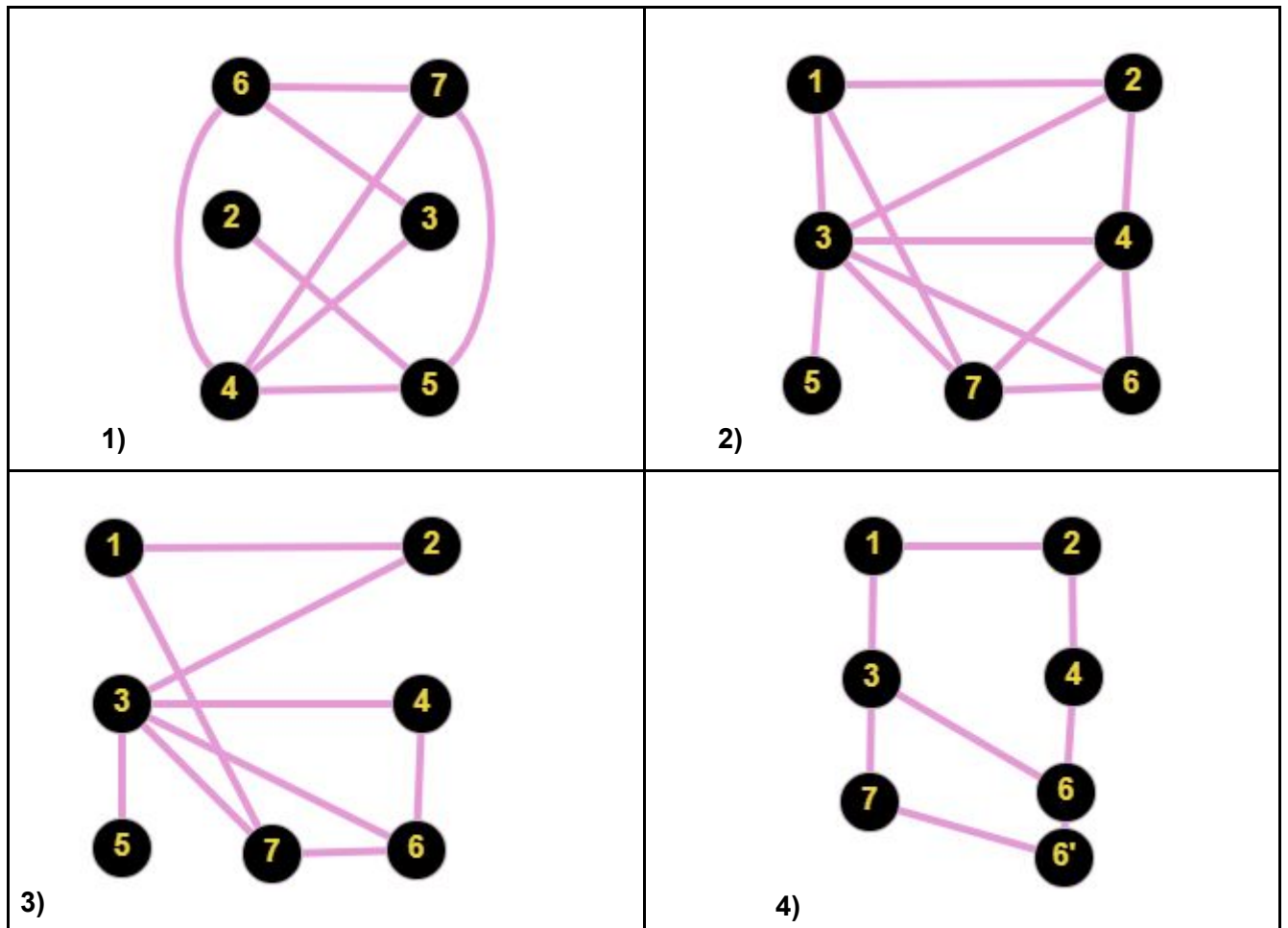
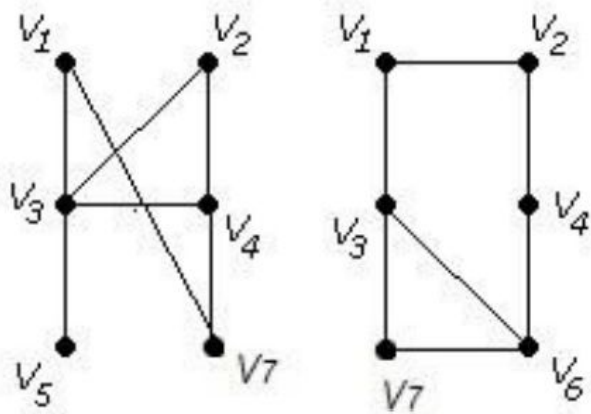
Перевірила:

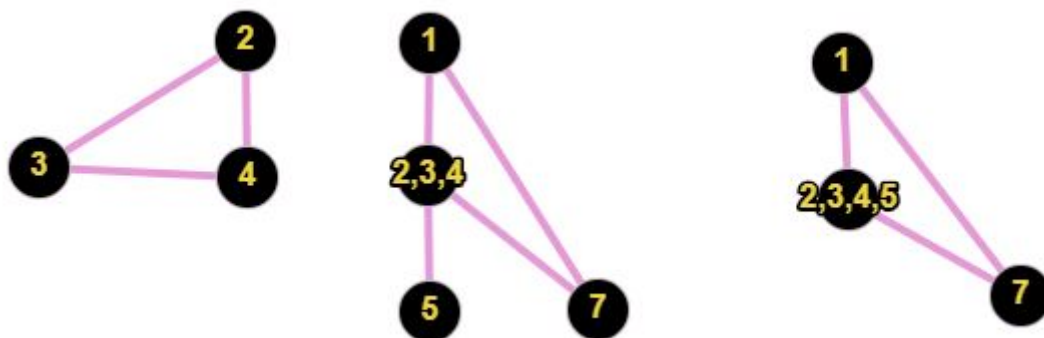
Мельникова Н. І.

Львів-2019

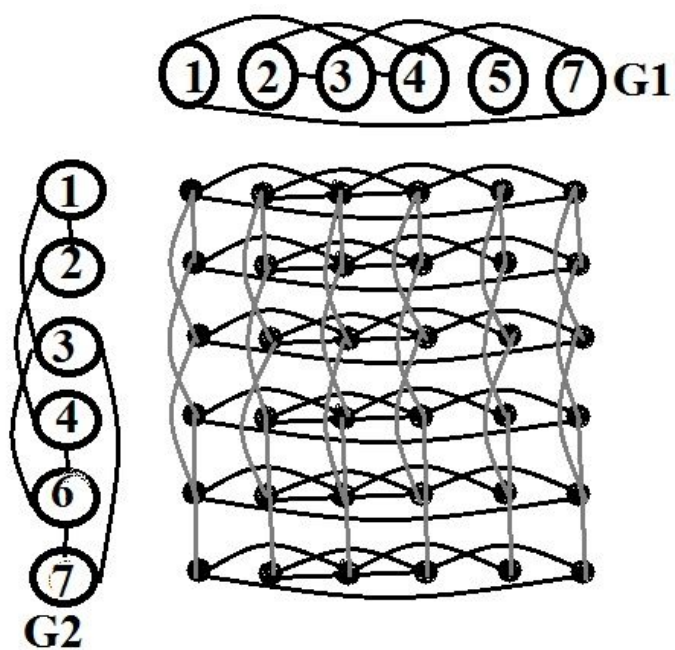
Завдання № 1

Виконати наступні операції над графами: 1) знайти доповнення до першого графу, 2) об'єднання графів, 3) кільцеву суму $G1$ та $G2$ ($G1+G2$), 4) розмножити вершину у другому графі, 5) виділити підграф A - що складається з 3-х вершин в $G1$ 6) добуток графів.





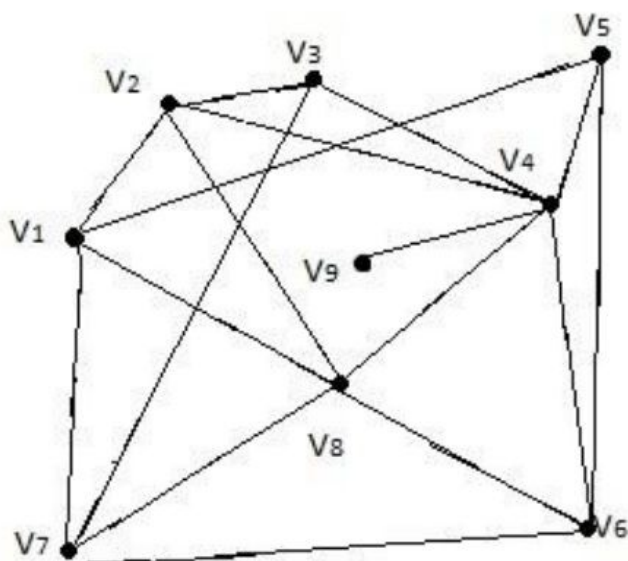
5)



6)

Завдання № 2

Скласти таблицю суміжності для орграфа



	1	2	3	4	5	6	7	8	9
1	-	1	0	0	1	0	1	0	0
2	1	-	1	1	0	0	0	1	0
3	0	1	-	1	0	0	1	0	0
4	0	1	1	-	1	1	0	0	1
5	1	0	0	1	-	1	0	0	0
6	0	0	0	1	1	-	1	1	0
7	1	0	1	0	0	1	-	1	0
8	0	1	0	0	0	1	1	-	0
9	0	0	0	1	0	0	0	0	-

Завдання № 3

Для графа з другого завдання знайти діаметр.

Відповідь: Діаметр = 3.

Завдання № 4

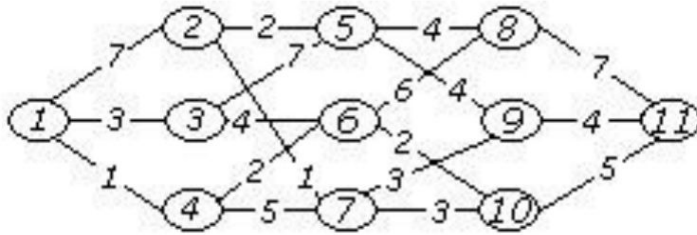
Для графа з другого завдання виконати обхід дерева вглиб.

Почнемо з довільної вершини V9

Стек	Вершина, яку додали	Присвоїли номер	Стек	Вершина, яку додали	Присвоїли номер
9	9	1	9,4,3,2,8,1,7	-	-
9,4	4	2	9,4,3,2,8,1	-	-
9,4,3	3	3	9,4,3,2,8	-	-
9,4,3,2	2	4	9,4,3,2	-	-
9,4,3,2,8	8	5	9,4,3	-	-
9,4,3,2,8,1	1	6	9,4	-	-
9,4,3,2,8,1,7	7	7	9	-	-
9,4,3,2,8,1,7,6	6	8	стек порожній	-	-
9,4,3,2,8,1,7,6,5	5	9			

Завдання № 5

Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.

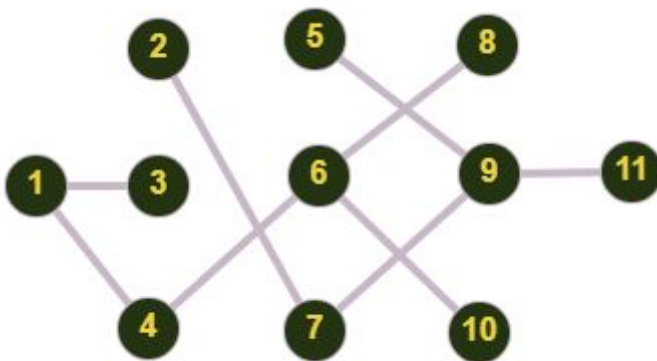


Метод Краскала:

1. V1+V4
2. V2+V7
3. V4+V6
4. V6+V10
5. V1+V3
6. V7+V9
7. V9+V5
8. V9+V11
9. V6+V8

Метод Прима:

1. V1+V4
2. V4+V6
3. V6+V10
4. V10+V7
5. V7+V2
6. V2+V5
7. V5+V9
8. V9+V11
9. V11+V8



Завдання № 6

Розв'язати задачу комівояжера для повного 8-ми вершинного графа методом «іди у найближчий», матриця вагів якого має вигляд

23)

	1	2	3	4	5	6	7	8
1	∞	4	6	5	1	5	6	5
2	4	∞	6	5	5	5	5	7
3	6	6	∞	1	2	3	2	5
4	5	5	1	∞	1	5	7	5
5	1	5	2	1	∞	5	5	6
6	5	5	3	5	5	∞	7	1
7	6	5	2	7	5	7	∞	2
8	5	7	5	5	6	1	2	∞

Почнемо з 1

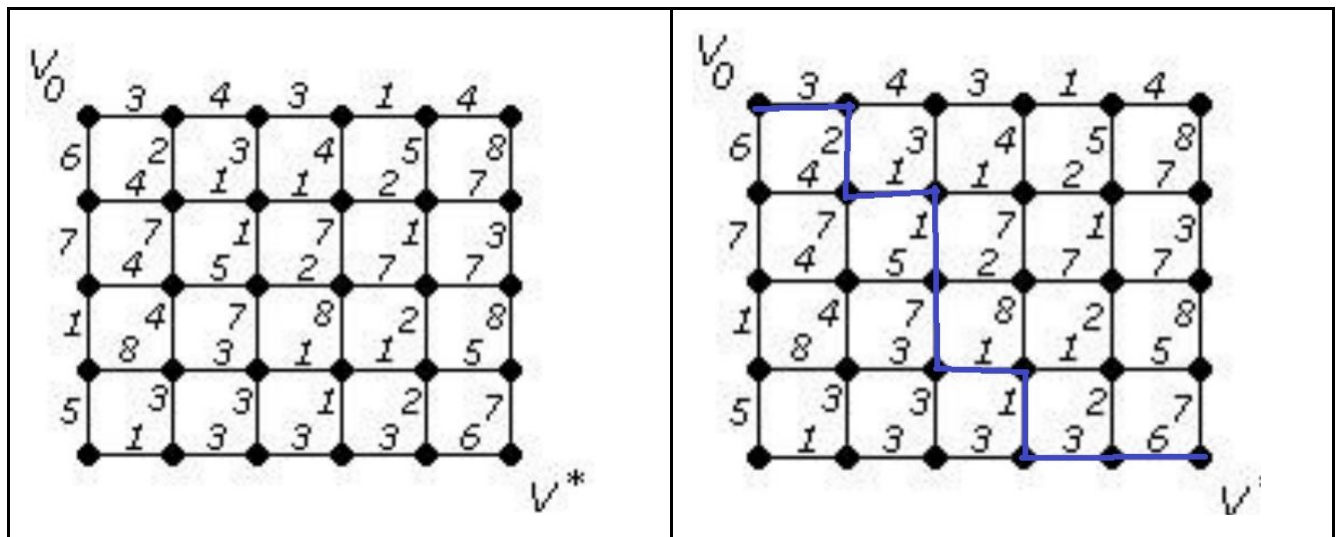
- 1) 1-5-4-3-7-8-6-2---1 d=12
- 2) 1-2-4-5-3-7-8-6---1 d=22
- 3) 1-4-3-5-2-6-8-7---1 d=27
- 4) 1-6-3-4-5-2-7-8---1 d=27
- 5) 1-8-6-3-4-5-2-7---1 d=27
- 6) 1-3-4-5-6-2-7-8---1 d=30
- 7) 1-7-3-4-5-2-6-8---1 d=26

Очевидно, що шляхом Комівояжера є: **1-5-4-3-7-8-6-2---1 d=12**

Завдання № 7

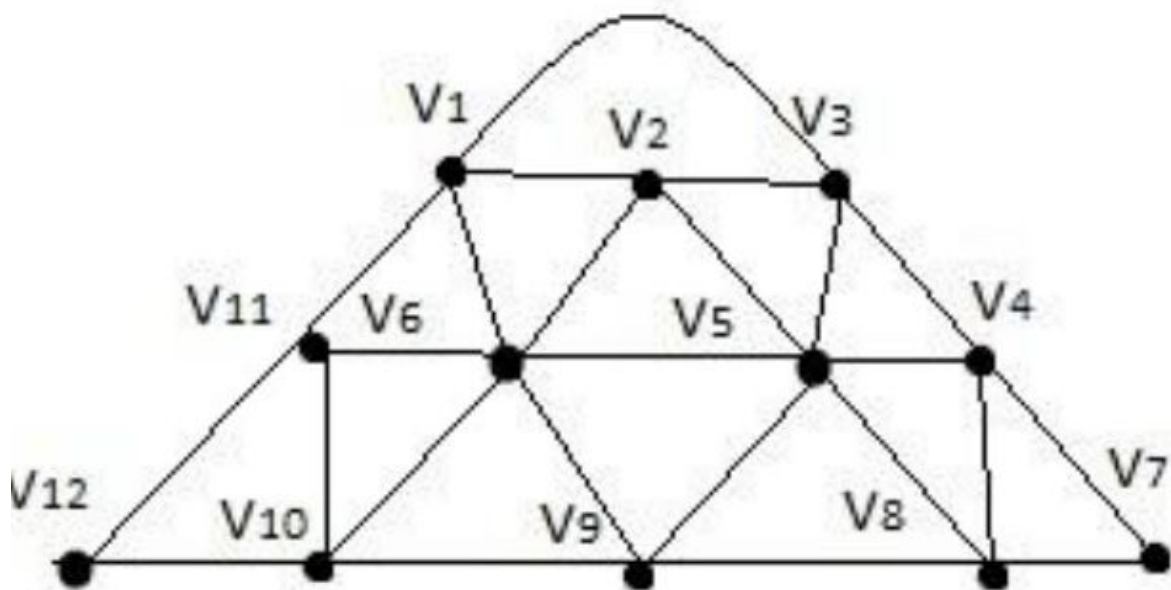
За допомогою алгоритму Дейкстри знайти найкоротший шлях у графі між парою вершин V_0 і V_* .

Алгоритм Дейкстри полягає у тому, щоб розглянути всі варіанти "шляху" до вершини і вибрати з них найоптимальніший (-> найкоротший)



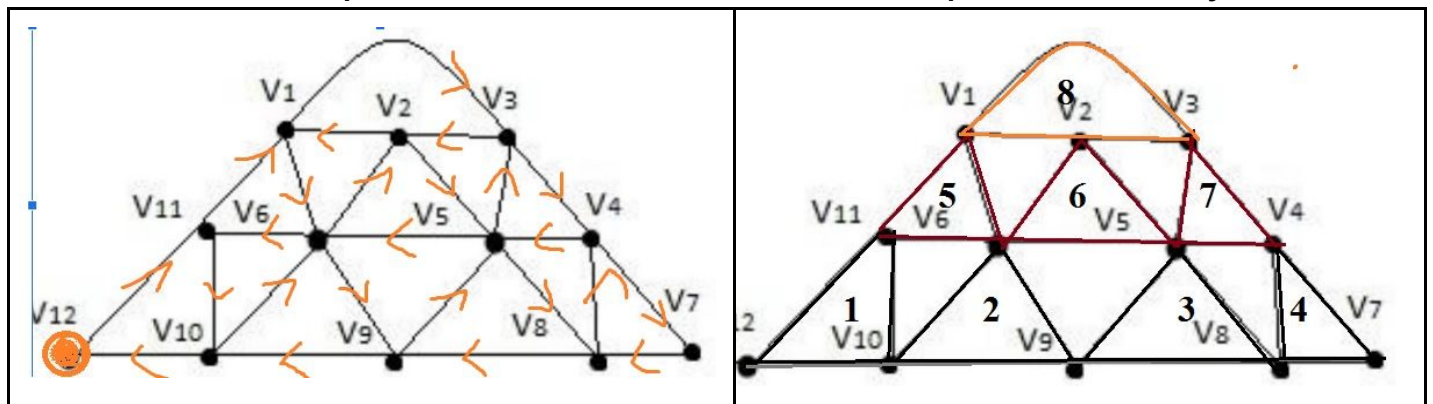
Завдання № 8

Знайти ейлеровий цикл в ейлеровому графі двома методами: а) Флері; б) елементарних циклів.



Почнемо з дов. вершини V_{12}

Знаходимо елементарні цикли і об'єднуємо їх в один



1-3-2-1-6-2-5-3-4-7-8-4-5-8-9-5-6-9-10-6-11-10-12

Завдання №9

Спростити формули (привести їх до скороченої ДНФ).

$$23. (x \vee \bar{y})(\bar{y} \vee \bar{z})$$

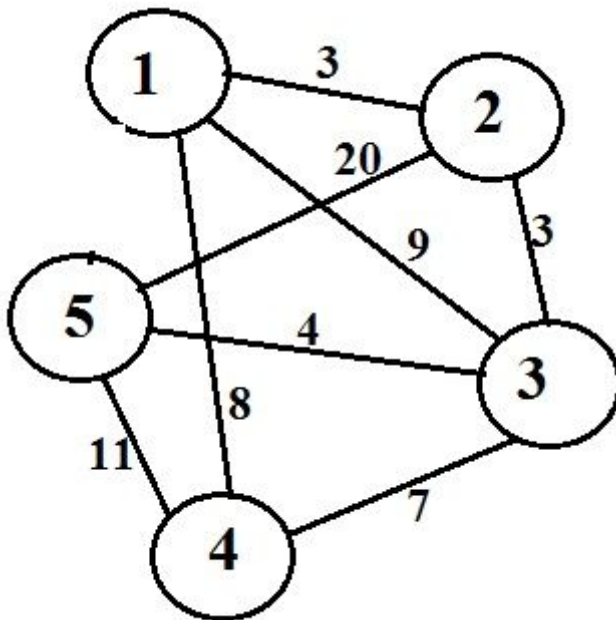
$$(x \vee \bar{y}) \wedge (\bar{y} \vee \bar{z}) = ((x \vee \bar{y}) \wedge \bar{y}) \vee ((x \vee \bar{y}) \wedge \bar{z}) = (x \vee \bar{y}) \vee (\bar{y} \vee \bar{y}) \vee (x \wedge \bar{z}) \vee (\bar{y} \wedge \bar{z})$$

Завдання №10

Напишіть алгоритм:

«Іди в найближчий» для розв'язання задачі комівояжера.

Для програмної реалізації візьмемо довільний граф:



Програма:

```
#include <stdlib.h>
#include <time.h>
#include <stdio.h>
#define SIZE 10

int wpchk(int w, int *wpts)
{
    int i=0;
    int flg=0;
    while(wpts[i]!=-1)
    {
```



```

        if(wpts[i]==w){flg=1;}
        i++;
    }
    if (flg==0) {return 0;} else return 1;
}

int main()
{
    srand( (unsigned)time( NULL ) );

    //int prices[10][10];
    int waypoint[11]={-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,1};
    int way[11]={-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1};
    int start=-1;
    int end=-1;
    int min;
    int imin;

    int prices[][10]={// 0 1 2 3 4 5 6 7 8 9
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //0
        { 0, 0, 2, 9, 8, 0, 0, 0, 0, 0}, //1
        { 0, 2, 0, 3, 0, 20,0, 0, 0, 0}, //2
        { 0, 9, 3, 0, 7, 4, 0, 0, 0, 0}, //3
        { 0, 8, 0, 7, 0, 11,0, 0, 0, 0}, //4
        { 0, 0, 20,4, 11,0, 0, 0, 0, 0}, //5
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //6
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //7
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //8
        { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0} //9
    };

    printf("Enter # of start location:");
    scanf("%i",&start);
    printf("Enter # of finish location:");
    scanf("%i",&end);
    waypoint[0]=start;
    int n=0;
    int w;
    while(waypoint[n]!=end)
    {
        min=0;
        w=waypoint[n];
        for(int i=0;i<SIZE;i++)
        {
            if(((min==0)||((prices[w][i]<min)&&(prices[w][i]>0))))&&wpchk(i,waypoint)==0)
            {min=prices[w][i];imin=i;}
        }
    }
}

```

```
        }
        n++;
        waypoint[n]=imin;
    }
    printf("\nThe way is:\n");
    int i=0;
    while(waypoint[i]!=-1)
    {
        printf("%i ",waypoint[i]);
        i++;
    }
}
```