جامعة المنارة

كلية: الهندسة

قسم: الهندسة المعلوماتية

اسم المقرر: الخوارزميات و بنى المعطيات2

رقم الجلسة (السادسة)

عنوان الجلسة

خوارزمية ديجسترا لإيجاد أقصر مسارات من مصدر أحادي في البيان (Dijkstra's Algorithm)

الفصل الدراسي الثاني

العام الدراسي 2023- 2024

الغاية من الجلسة

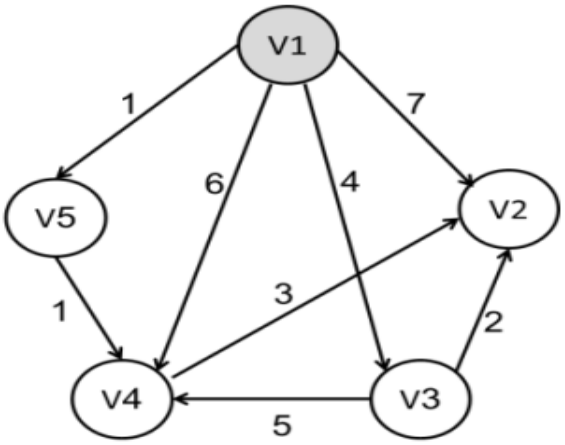
- ✓ تطبيق خوارزمية ديجسترا لإيجاد أقصر مسارات من رأس مصدر أحادي في البيان .
- ✓ تنفيذ الكود الخاص بخوارزمية ديجسترا .

خوارزمية ديجسترا :

تستخدم خوارزمية ديجسترا من أجل إيجاد أقصر مسارات من رأس مصدر أحادي إلى جميع الرؤوس الأخرى في بيان موزون .

تمرين:

طبق خوارزمية ديجسترا لإيجاد أقصر مسارات من الرأس المصدر (v1) إلى جميع الرؤوس الأخرى في البيان :



1- نختار الرأس 1 ليكون الرأس المصدر و نعرف المصفوفة length التي تمثل أوزان المسارات التي تصل كل رأس في البيان بالرأس المصدر (1) بقيم مصفوفة التجاور للرأس 1 ، كما نهيئ المصفوفة touch بالقيمة 1 دلالة على الرأس 1 .

Touch Array			
2	3	4	5
1	1	1	1

Length Array			
2	3	4	5
7	4	6	1

2- في كل تكرار نختار من المصفوفة length الرأس V_{near} ذي المسار الأقصر عن الرأس V_1 ، و من أجل كل رأس V_i مجاور للرأس V_{near} و لم تتم زيارته بعد، يتم حساب المجموع التالي : وزن الحرف (V_{near}, V_i) + وزن المسار الذي يربط V_{near} مع الرأس V_1 (length[Vnear])، و نقارن هذا المجموع مع وزن المسار الحالي للرأس V_i

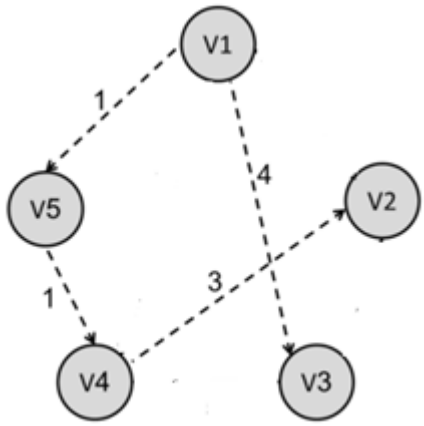
(length[Vi])، فإذا كان أقل منه يتم تعديل المصفوفة length بالأوزان الجديدة ،ويقابل ذلك تعديل في قيم المصفوفة touch الموافقة، و نكرر هذه العمليات حتى تتم زيارة جميع الرؤوس كمايلي :

رقم التكرار	Touch Array			
	2	3	4	5
1	1	1	1	1
2	1	1	5	1
3	4	1	5	1
4	4	1	5	1

رقم التكرار	Length Array			
	2	3	4	5
1	7	4	6	1
2	7	4	2	-1
3	5	4	-1	-1
4	5	-1	-1	-1
stop	-1	-1	-1	-1

MIN	vnear
1	5
2	4
4	3
5	2

المسارات الأقصر
(1,5)
(5,4)
(1,3)
(4,2)



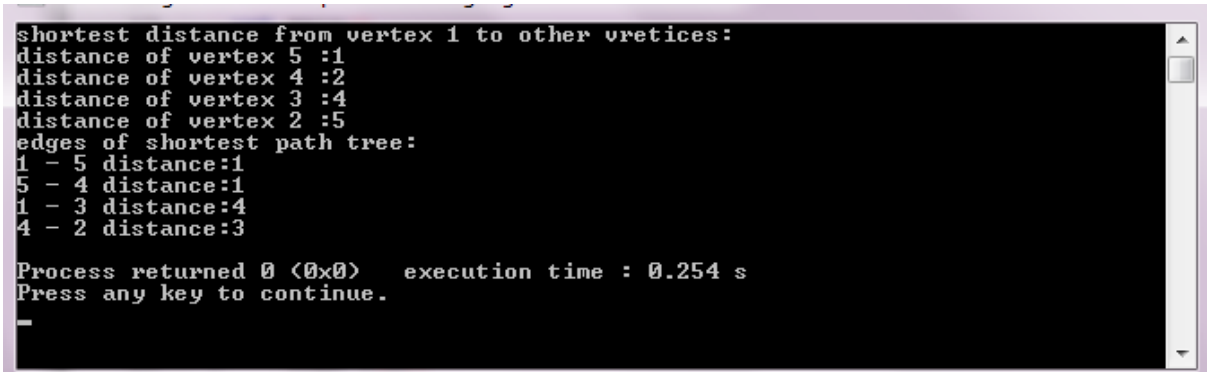
البرنامج الخاص بخوارزمية ديجسترا:

```
#include <iostream>
using namespace std;
#define N 5
#define INF 1000
int f_size = 0;
 struct edge
  {
 int weight;
 int v1;
 int v2;
  };

void dijkstra( int(*W)[N + 1], edge* F);
```

```
int main()
{
int W[N+1][N+1]={{0,0,0,0,0,0},
                 {0,0,7,4,6,1},
                 {0,INF,0,INF,INF,INF},
                 {0,INF,2,0,5,INF},
                 {0,INF,3,INF,0,INF},
                 {0,INF,INF,INF,1,0}};
 edge F[N];
    cout<<"shortest distance from vertex 1 to other
vretices:\n";

    dijkstra(W, F);
    cout<<"edges of shortest path tree:\n";

 for (int i = 0; i < f_size; i++)
 {
     cout<< "distance:"<< F[i].v1<<" "<< F[i].v2<<"
"<<W[F[i].v1][F[i].v2]<<endl;
 }
}

void dijkstra( int (*W)[N+1], edge *F)
{
 int vnear;
 edge e;
 int touch[N + 1];
 int length[N + 1];
 for (int i = 2; i < N+1; i++)
 {
     touch[i] = 1;
     length[i] = W[1][i];
 }
 for(int i=2;i<N+1;i++)
 {
     int min = INF;
     //find vertex with minimum length
     for (int i=2; i < N+1; i++)
     {
          if (0 < length[i] && length[i]< min)
          {
               min = length[i];
               vnear = i;
          }
```

```
      }
      //edge from touch[vnear] to vnear

      e.v1 = touch[vnear];
      e.v2 = vnear;
      F[f_size] = e;
      f_size++;
      for (int i = 2; i <N+1; i++)
      {
            if (length[vnear] + W[vnear][i] < length[i])
            {
                  length[i] = length[vnear] + W[vnear][i];
                  touch[i] = vnear;
            }
      }
      cout<<"distance of vertex "<< vnear<<" : "<<
length[vnear]<<"\n";
      //vnear is visited
      length[vnear] = -1;
 }
}
```

خرج البرنامج :



تمرين غير محلول:

طبق خوارزمية ديجسترا لإيجاد أقصر مسارات من الرأس v1 إلى جميع الرؤوس الأخرى في البيان التالي: