

جامعة المنارة

كلية: الهندسة

قسم: الهندسة المعلوماتية

اسم المقرر: الخوارزميات وبنى المعطيات 2

رقم الجلسة (السادسة)

عنوان الجلسة

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



الفصل الدراسي الأول
العام الدراسي 2023-2024

الغاية من الجلسة

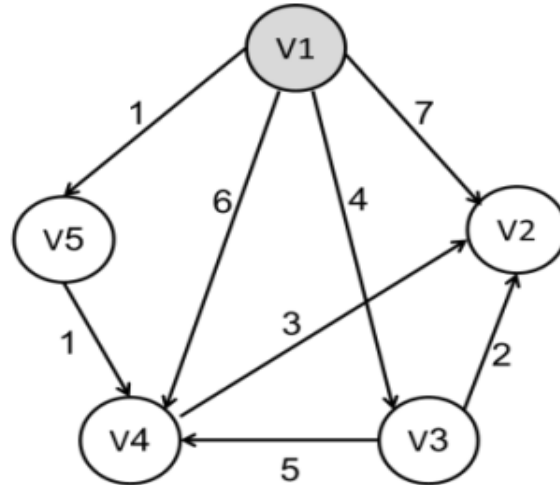
- ✓ تطبيق خوارزمية ديجسترا لإيجاد أقصر مسار في البيان .
- ✓ تنفيذ الكود الخاص بخوارزمية ديجسترا .

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 رأس البداية و نرئ المصفوفة length بقيم مصفوفة التجاور للرأس 1 ، كما نرئ المصفوفة touch بالقيمة 1 دلالة على الرأس 1 .

Touch Array				Length Array			
2	3	4	5	2	3	4	5
1	1	1	1	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} ولم تتم زيارته بعد، يتم حساب المجموع التالي : وزن الحرف (V_{near}, V_i) + وزن المسار الذي يربط V_{near} مع الرأس V_1 ، ونقارن هذا المجموع مع المسار الحالي للرأس V_i ، فإذا كان أقل منه يتم

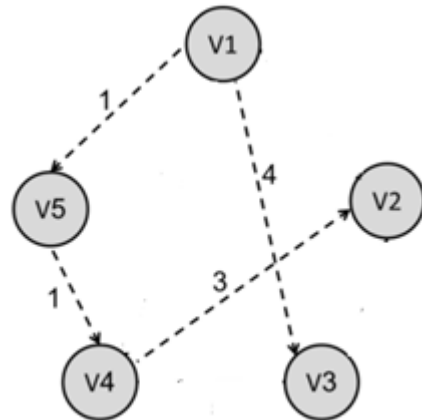
تعديل المصفوفة length بالأوزان الجديدة، ويقابل ذلك تعديل في قيم المصفوفة touch الموافقة، و نكرر هذه العمليات حتى تتم زيارة جميع الرؤوس كمايلي :

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;
struct edge
{
int weight;
int v1;
int v2;
};

#define N 5
#define INF 1000

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

```

int f_index = -1;

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];
    printf("shortest distance from vertex 1 to other
vretices:\n");

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

for (int i = 0; i <= f_index; i++)
{
    printf("%d - %d distance:%d\n", F[i].v1,
F[i].v2,W[F[i].v1][F[i].v2]);
}

}

void dijkstra(int n, int (*W)[N+1], edge *F)
{
int vnear;
edge e;

int touch[N + 1];
int length[N + 1];

for (int i = 2; i <= n; i++)
{
    touch[i] = 1;
    length[i] = W[1][i];
}

while (1)

```

```
{

    int min = INF;
    //find vertex with minimum length
    for (int i=2; i <= n; 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_index] = e;

    for (int i = 2; i <= n; i++)
    {
        if (length[vnear] + W[vnear][i] < length[i])
        {
            length[i] = length[vnear] + W[vnear][i];
            touch[i] = vnear;
        }
    }
    printf("distance of vertex %d :%d\n", vnear,
length[vnear]);
    //vnear is visited
    length[vnear] = -1;

    if (f_index == n - 2)
        break;
}

}
```

خرج البرنامج :

```

shortest distance from vertex 1 to other vertices:
distance of vertex 5 :1
distance of vertex 4 :2
distance of vertex 3 :4
distance of vertex 2 :5
edges of shortest path tree:
1 - 5 distance:1
5 - 4 distance:1
1 - 3 distance:4
4 - 2 distance:3

Process returned 0 (0x0)   execution time : 0.254 s
Press any key to continue.

```

تمرين غير محلول:

طبق خوارزمية ديجسترا لإيجاد أقصر مسارات من الرأس v_1 إلى جميع الرؤوس الأخرى في البيان التالي:

