

[Dashboard](#) / [My courses](#) / [Graph Theory-HK3-0405](#) / [Tuần 6 - 7 - Đường đi ngắn nhất trên đồ thị](#) / [Bài tập 3 - Thuật toán Moore - Dijkstra \(đường đi\)](#)

| | |
|---------------------|---|
| Started on | Saturday, 21 June 2025, 4:03 PM |
| State | Finished |
| Completed on | Saturday, 21 June 2025, 4:25 PM |
| Time taken | 22 mins 12 secs |
| Marks | 2.00/2.00 |
| Grade | 10.00 out of 10.00 (100%) |

Question 1

Correct

Mark 1.00 out of 1.00

Viết chương trình đọc một **đơn đồ thị có hướng, có trọng số không âm** từ bàn phím, tìm đường đi ngắn nhất từ đỉnh s đến đỉnh t (s và t cũng được đọc từ bàn phím).

Đầu vào (Input)

Dữ liệu đầu vào được nhập từ bàn phím với định dạng:

- Dòng đầu tiên chứa 2 số nguyên n và m ($1 \leq n < 100; 0 \leq m < 500$)
- m dòng tiếp theo mỗi dòng chứa 3 số nguyên u, v, w mô tả cung (u, v) có trọng số w ($0 \leq w \leq 100$).
- Dòng cuối cùng chứa 2 số nguyên s và t.

Đầu ra (Output)

- In đường đi ngắn nhất từ s đến t theo mẫu:

s -> u1 -> u2 -> ... -> t

- Xem thêm ví dụ bên dưới.

Gợi ý

- Lần ngược theo p[u] để có được đường đi ngắn nhất.

For example:

| Input | Result |
|--|-------------|
| 3 3 1 2 3 2 3 5 1 3 10 1 3 | 1 -> 2 -> 3 |

Answer: (penalty regime: 10, 20, ... %)

```
1 #include <stdio.h>
2 #define MAX_N 100
3 #define NO_EDGE -1
4 #define oo 999999
5
6 typedef struct
7 {
8     int n, m;
9     int W[MAX_N][MAX_N];
10 } Graph;
11
12 // ham khoi tao
13 void init_graph(Graph *pG, int n)
14 {
15     pG->n = n;
16     pG->m = 0;
17     for (int u = 1; u <= pG->n; u++)
18         for (int v = 1; v <= pG->n; v++)
19             pG->W[u][v] = NO_EDGE;
20 }
21
22 // them cung vao do thi
```

| | Input | Expected | Got | |
|---|--|-------------|-------------|---|
| ✓ | 3 3 1 2 3 2 3 5 1 3 10 1 3 | 1 -> 2 -> 3 | 1 -> 2 -> 3 | ✓ |
| ✓ | 3 3 1 3 5 3 2 3 1 2 10 1 2 | 1 -> 3 -> 2 | 1 -> 3 -> 2 | ✓ |

| | Input | Expected | Got | |
|---|---|------------------|------------------|---|
| ✓ | 6 9 1 2 7 1 3 9 1 5 14 2 3 10 2 4 15 3 4 11 3 5 2 4 6 6 5 6 9 1 5 | 1 -> 3 -> 5 | 1 -> 3 -> 5 | ✓ |
| ✓ | 6 9 1 2 7 1 3 9 1 5 14 2 3 10 2 4 15 3 4 11 3 5 2 4 6 6 5 6 9 2 6 | 2 -> 3 -> 5 -> 6 | 2 -> 3 -> 5 -> 6 | ✓ |
| ✓ | 6 9 1 2 7 1 3 9 1 5 14 2 3 10 2 4 15 3 4 11 3 5 3 4 6 6 6 5 9 2 5 | 2 -> 3 -> 5 | 2 -> 3 -> 5 | ✓ |

Passed all tests! ✓

Question author's solution (C):

```

1 #include <stdio.h>
2
3 #define MAXN 100
4 #define oo 999999
5 #define NO_EDGE -1
6
7
8 typedef struct {
9     int n, m;
10    int W[MAXN][MAXN];
11 } Graph;
12
13 void init_graph(Graph *pG, int n) {
14     pG->n = n;
15     pG->m = 0;
16     for (int u = 1; u <= n; u++)
17         for (int v = 1; v <= n; v++)
18             pG->W[u][v] = NO_EDGE;
19 }
20
21 void add_edge(Graph *pG, int u, int v, int w) {
22     pG->W[u][v] = w;

```

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Viết chương trình đọc một **đơn đồ thị vô hướng, có trọng số không âm** từ bàn phím, tìm đường đi ngắn nhất từ đỉnh s đến đỉnh t (s và t cũng được đọc từ bàn phím).

Đầu vào (Input)

Dữ liệu đầu vào được nhập từ bàn phím với định dạng:

- Dòng đầu tiên chứa 2 số nguyên n và m ($1 \leq n < 100; 0 \leq m < 500$)
- m dòng tiếp theo mỗi dòng chứa 3 số nguyên u, v, w mô tả cung (u, v) có trọng số w ($0 \leq w \leq 100$).
- Dòng cuối cùng chứa 2 số nguyên s và t.

Đầu ra (Output)

- In đường đi ngắn nhất từ s đến t theo mẫu:

s -> u1 -> u2 -> ... -> t

- Xem thêm ví dụ bên dưới.

Gợi ý

- Lần ngược theo p[u] để có được đường đi ngắn nhất.

For example:

| Input | Result |
|--|-------------|
| 3 3 1 2 3 2 3 5 1 3 10 3 1 | 3 -> 2 -> 1 |

Answer: (penalty regime: 10, 20, ... %)

```

1 #include <stdio.h>
2 #define MAX_N 100
3 #define NO_EDGE -1
4 #define oo 999999
5
6 typedef struct
7 {
8     int n, m;
9     int W[MAX_N][MAX_N];
10 } Graph;
11
12 // hàm khởi tạo
13 void init_graph(Graph *pG, int n)
14 {
15     pG->n = n;
16     pG->m = 0;
17     for (int u = 1; u <= pG->n; u++)
18         for (int v = 1; v <= pG->n; v++)
19             pG->W[u][v] = NO_EDGE;
20 }
21
22 // thêm cung vào đồ thị

```

| | Input | Expected | Got | |
|---|--|-------------|-------------|---|
| ✓ | 3 3 1 2 3 2 3 5 1 3 10 3 1 | 3 -> 2 -> 1 | 3 -> 2 -> 1 | ✓ |
| ✓ | 3 3 1 3 5 3 2 3 1 2 10 2 1 | 2 -> 3 -> 1 | 2 -> 3 -> 1 | ✓ |

| | Input | Expected | Got | |
|---|---|------------------|------------------|---|
| ✓ | 6 9 1 2 7 1 3 9 1 5 14 2 3 10 2 4 15 3 4 11 3 5 2 4 6 6 5 6 9 6 2 | 6 -> 4 -> 2 | 6 -> 4 -> 2 | ✓ |
| ✓ | 6 9 1 2 7 1 3 9 1 5 14 2 3 10 2 4 15 3 4 11 3 5 2 4 6 6 5 6 9 6 1 | 6 -> 5 -> 3 -> 1 | 6 -> 5 -> 3 -> 1 | ✓ |
| ✓ | 6 9 1 2 7 1 3 9 1 5 14 2 3 10 2 4 15 3 4 11 3 5 3 4 6 6 6 5 9 5 2 | 5 -> 3 -> 2 | 5 -> 3 -> 2 | ✓ |

Passed all tests! ✓

Question author's solution (C):

```

1 #include <stdio.h>
2
3 #define MAXN 100
4 #define oo 999999
5 #define NO_EDGE -1
6
7
8 typedef struct {
9     int n, m;
10    int W[MAXN][MAXN];
11 } Graph;
12
13 void init_graph(Graph *pG, int n) {
14     pG->n = n;
15     pG->m = 0;
16     for (int u = 1; u <= n; u++)
17         for (int v = 1; v <= n; v++)
18             pG->W[u][v] = NO_EDGE;
19 }
20
21 void add_edge(Graph *pG, int u, int v, int w) {
22     pG->W[u][v] = w;

```

Correct

Marks for this submission: 1.00/1.00.

[← Bài tập 2 - Thuật toán Moore - Dijkstra \(chiều dài\)](#)

Jump to...

[Bài tập 4* - Mê cung số ►](#)