

[Dashboard](#) / [My courses](#) / [Graph Theory-HK3-0405](#) / [Tuần 6 - 7 - Đường đi ngắn nhất trên đồ thị](#) / [Bài tập 1 - Thuật toán Moore - Dijkstra \(pi và p\)](#)

Started on	Friday, 20 June 2025, 4:55 PM
State	Finished
Completed on	Saturday, 21 June 2025, 9:27 AM
Time taken	16 hours 32 mins
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. Cài đặt thuật toán Moore – Dijkstra để tìm (các) đường đi ngắn nhất từ đỉnh 1 đến các đỉnh còn lại. In các thông tin $pi[u]$ và $p[u]$ của các đỉnh ra màn hình.

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 ($0 \leq w \leq 100$).

Đầu ra (Output)

- In ra màn hình $pi[u]$ và $p[u]$ của các đỉnh theo mẫu:

```
pi[1] = 0, pi[1] = -1
pi[2] = 3, pi[2] = 1
...
```

Xem thêm ví dụ bên dưới.

Chú ý

- Để chạy thử chương trình, bạn nên tạo một tập tin **dt.txt** chứa đồ thị cần kiểm tra.
- Thêm dòng `freopen("dt.txt", "r", stdin);` vào ngay sau hàm `main()`. Khi nộp bài, nhớ gỡ bỏ dòng này ra.
- Có thể sử dụng đoạn chương trình đọc dữ liệu mẫu sau đây:

```
freopen("dt.txt", "r", stdin); //Khi nộp bài nhớ bỏ dòng này.
Graph G;
int n, m, u, v, w, e;
scanf("%d%d", &n, &m);
init_graph(&G, n);

for (e = 0; e < m; e++) {
    scanf("%d%d%d", &u, &v, &w);
    add_edge(&G, u, v, w);
}
```

For example:

Input	Result
3 4	$pi[1] = 0, p[1] = -1$
1 2 9	$pi[2] = 4, p[2] = 3$
2 3 5	$pi[3] = 4, p[3] = 1$
1 3 4	
3 2 0	

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 4 1 2 9 2 3 5 1 3 4 3 2 0	pi[1] = 0, p[1] = -1 pi[2] = 4, p[2] = 3 pi[3] = 4, p[3] = 1	pi[1] = 0, p[1] = -1 pi[2] = 4, p[2] = 3 pi[3] = 4, p[3] = 1	✓
✓	3 4 1 2 5 2 3 1 1 3 10 3 1 4	pi[1] = 0, p[1] = -1 pi[2] = 5, p[2] = 1 pi[3] = 6, p[3] = 2	pi[1] = 0, p[1] = -1 pi[2] = 5, p[2] = 1 pi[3] = 6, p[3] = 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	pi[1] = 0, p[1] = -1 pi[2] = 7, p[2] = 1 pi[3] = 9, p[3] = 1 pi[4] = 20, p[4] = 3 pi[5] = 11, p[5] = 3 pi[6] = 20, p[6] = 5	pi[1] = 0, p[1] = -1 pi[2] = 7, p[2] = 1 pi[3] = 9, p[3] = 1 pi[4] = 20, p[4] = 3 pi[5] = 11, p[5] = 3 pi[6] = 20, p[6] = 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. Cài đặt thuật toán Moore – Dijkstra để tìm (các) đường đi ngắn nhất từ đỉnh 1 đến các đỉnh còn lại. In các thông tin $pi[u]$ và $p[u]$ của các đỉnh ra màn hình.

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 ($0 \leq w \leq 100$).

Đầu ra (Output)

- In ra màn hình $pi[u]$ và $p[u]$ của các đỉnh theo mẫu:

```
pi[1] = 0, p[1] = -1
pi[2] = 3, p[2] = 1
...
```

Xem thêm ví dụ bên dưới.

Chú ý

- Để chạy thử chương trình, bạn nên tạo một tập tin **dt.txt** chứa đồ thị cần kiểm tra.
- Thêm dòng `freopen("dt.txt", "r", stdin);` vào ngay sau hàm `main()`. Khi nộp bài, nhớ gỡ bỏ dòng này ra.
- Có thể sử dụng đoạn chương trình đọc dữ liệu mẫu sau đây:

```
freopen("dt.txt", "r", stdin); //Khi nộp bài nhớ bỏ dòng này.
Graph G;
int n, m, u, v, w, e;
scanf("%d%d", &n, &m);
init_graph(&G, n);

for (e = 0; e < m; e++) {
    scanf("%d%d%d", &u, &v, &w);
    add_edge(&G, u, v, w);
}
```

For example:

Input	Result
4 5	$pi[1] = 0, p[1] = -1$
1 2 10	$pi[2] = 9, p[2] = 4$
3 2 6	$pi[3] = 4, p[3] = 1$
3 1 4	$pi[4] = 4, p[4] = 3$
3 4 0	
4 2 5	

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	
✓	4 5 1 2 10 3 2 6 3 1 4 3 4 0 4 2 5	pi[1] = 0, p[1] = -1 pi[2] = 9, p[2] = 4 pi[3] = 4, p[3] = 1 pi[4] = 4, p[4] = 3	pi[1] = 0, p[1] = -1 pi[2] = 9, p[2] = 4 pi[3] = 4, p[3] = 1 pi[4] = 4, p[4] = 3	✓
✓	3 3 3 1 8 1 2 4 3 2 1	pi[1] = 0, p[1] = -1 pi[2] = 4, p[2] = 1 pi[3] = 5, p[3] = 2	pi[1] = 0, p[1] = -1 pi[2] = 4, p[2] = 1 pi[3] = 5, p[3] = 2	✓
✓	6 9 1 2 7 1 3 9 5 1 14 2 3 10 4 4 15 3 4 11 3 5 2 4 6 6 6 5 12	pi[1] = 0, p[1] = -1 pi[2] = 7, p[2] = 1 pi[3] = 9, p[3] = 1 pi[4] = 20, p[4] = 3 pi[5] = 11, p[5] = 3 pi[6] = 23, p[6] = 5	pi[1] = 0, p[1] = -1 pi[2] = 7, p[2] = 1 pi[3] = 9, p[3] = 1 pi[4] = 20, p[4] = 3 pi[5] = 11, p[5] = 3 pi[6] = 23, p[6] = 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.

◀ * 7. Moore - Dijkstra trực tiếp trên đồ thị (random)

Jump to...

Bài tập 2 - Thuật toán Moore - Dijkstra (chiều dài) ▶