

[Dashboard](#) / [My courses](#) / [Graph Theory-HK3-0405](#) / [Tuần 6 - 7 - Đường đi ngắn nhất trên đồ thị](#) / [Bài tập 4* - Mê cung số](#)

Started on	Tuesday, 24 June 2025, 5:49 PM
State	Finished
Completed on	Tuesday, 24 June 2025, 5:51 PM
Time taken	2 mins 5 secs
Marks	1.00/1.00
Grade	10.00 out of 10.00 (100%)



Question 1

Correct

Mark 1.00 out of 1.00

Mê cung số (number maze)

Cho một mê cung số được biểu diễn bằng một mảng 2 chiều chứa các con số từ 0 đến 9 (xem hình bên dưới).

0	3	1	2	9
7	3	4	9	9
1	7	5	5	3
2	3	4	2	5

Một con robot được đặt tại **góc trên bên trái** của mê cung và muốn đi đến góc dưới bên phải của mê cung. Con robot có thể đi lên, xuống, qua trái và qua phải 1 ô. Chi phí để đi đến một ô bằng với con số bên trong ô đó.

Hãy tìm cách giúp con robot đi đến ô **góc dưới phải** sao cho tổng chi phí thấp nhất.

Đường đi có chi phí thấp nhất cho ví dụ này là 24.

Đầu vào (Input)

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

- Dòng đầu chứa 2 số nguyên M N (M: số hàng, N: số cột)
- M dòng tiếp theo mô tả các số trong mê cung

Đầu ra (Output)

- In ra màn hình chi phí thấp nhất để con robot đi từ góc trên bên trái về góc dưới bên phải.
- Trong ví dụ trên, cần in ra màn hình: 24.

Xem thêm các ví dụ bên dưới.

Gợi ý

- Mô hình hoá bài toán về đồ thị có hướng
 - Đỉnh ~ ô
 - Cung ~ hai ô cạnh nhau
 - Trọng số cung (u, v) = giá trị của ô tương ứng với đỉnh v.
- Xem tài liệu thực hành để biết cách đặt tên cho các ô.

For example:

Input	Result
4 5 0 3 1 2 9 7 3 4 9 9 1 7 5 5 3 2 3 4 2 5	24
3 3 1 2 3 2 4 4 1 4 10	17
4 3 1 2 2 2 1 4 4 2 1 8 4 10	16

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

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #define MAX_N 1000
4 #define oo 1000000
5
6 int n, m; // kích thước ma trận
7 int a[100][100];
8 int pi[MAX_N]; // chiều dài từ đỉnh s đến u
9 int p[MAX_N]; // đỉnh trước của u
10 int mark[MAX_N];
11
12 int dx[] = {-1, 1, 0, 0}, dy[] = {0, 0, -1, 1}; // trên, dưới, trái phải

```



```

13
14 void Dijkstra(int s)
15 {
16     int N = n * m; // diện tích của ma trận n hàng m cột
17     for (int i = 0; i < N; i++)
18     {
19         pi[i] = oo;
20         mark[i] = 0;
21         p[i] = -1;
22     }
23     pi[s] = 0; // chỉ phí bắt đầu bằng giá trị 0,0
24     for (int it = 0; it < N; it++)
25     {
26         int u = -1, mn = oo;
27         for (int i = 0; i < N; i++)
28             if (!mark[i] && pi[i] < mn)
29             {
30                 mn = pi[i];
31                 u = i;
32             }
33         if (u < 0) // nếu đã duyệt xong hết rồi thì break
34             break;
35         mark[u] = 1; // đánh dấu đã duyệt
36         int ux = u / m; // chỉ số hàng x
37         int uy = u % m; // chỉ số cột y
38         for (int k = 0; k < 4; k++)
39         { // dò tìm all ô bên cạnh chưa duyệt
40             int Vx = ux + dx[k], Vy = uy + dy[k];
41             // if (Vx >= 0 && Vx < n && Vy >= 0 && Vy < m)
42             //     continue;
43             if (Vx < 0 || Vx >= n || Vy < 0 || Vy >= m)
44                 continue;
45
46             int v = Vx * m + Vy; // địa chỉ ô thứ v
47             int w = a[Vx][Vy]; // giá trị của ô đó
48             if (!mark[v] && pi[u] + w < pi[v])
49             {
50                 pi[v] = pi[u] + w;
51                 p[v] = u;
52             }
53         }
54     }
55 }
56
57 int main()
58 {
59     scanf("%d%d", &n, &m);
60     for (int i = 0; i < n; i++)
61     {
62         for (int j = 0; j < m; j++)
63         {
64             scanf("%d", &a[i][j]);
65         }
66     }
67
68     Dijkstra(0);
69     int t = (n - 1) * m + (m - 1);
70     printf("%d\n", pi[t]);
71
72     return 0;
73 }

```

	Input	Expected	Got	
✓	4 5 0 3 1 2 9 7 3 4 9 9 1 7 5 5 3 2 3 4 2 5	24	24	✓
✓	3 3 1 2 3 2 4 4 1 4 10	17	17	✓

	Input	Expected	Got	
✓	4 3 1 2 2 2 1 4 4 2 1 8 4 10	16	16	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[◀ Bài tập 3 - Thuật toán Moore - Dijkstra \(đường đi\)](#)

Jump to...

[Bài tập 5* - Ô kiêu \(Ngưu Lang - Chức Nữ\) ▶](#)