



Problem Solving (C64)

هذا البرنامج التدريبي مُصاغ بعناية لتمكين المتدربين من تطوير قدراتهم الفكرية على غرار المبرمجين المحترفين، والتعاون بكفاءة ضمن فريق محترف في شركة "جمال تك" أو أي مؤسسة متعددة الجنسيات أخرى. نظرًا لأهمية اللغة الإنجليزية في بيئة العمل العالمية، يتم تقديم المحتوى التدريبي بالإنجليزية. لا يشترط إتقان اللغة بشكل كامل، لكن من الضروري امتلاك القدرة الكافية لفهم المتطلبات وتنفيذها بشكل فعال. يُمكن للمتدربين استخدام مترجم جوجل أو الاستعانة بـ "شات جي بي تي" للتغلب على أية عقبات لغوية، المهم هو الفهم الدقيق للمطلوب وتحقيقه بنجاح.

لتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمارين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق في نهاية الملف.

قد يتضمن الحل كودًا برمجيًا غير مفسر بعد، والغرض من ذلك هو تشجيعك على محاولة فهم الأكواد البرمجية الجديدة التي لم تتعرض لها من قبل. هذه المهارة ضرورية في سوق العمل، حيث تتطور لغات البرمجة باستمرار ويظهر كل يوم لغات جديدة. ستواجه دائمًا أكوادًا لم تدرسها من قبل، ومن المهم أن تكون قادرًا على فهمها بنفسك دون الحاجة إلى دراسة مسبقة. يمكنك الاستعانة بمحرك البحث جوجل، أو استخدام ChatGPT، أو حتى اللجوء لأصدقائك للمساعدة. الهدف الأساسي هو أن تصل إلى فهم معنى كل كود بأي طريقة ممكنة لتتمكن من إيجاد موقعك في سوق العمل.

إن وجود كود برمجي غير مفسر يشكل تحديًا يتوجب عليك إيجاد حل له. هذا النوع من التدريبات يعد جزءًا أساسيًا من تدريبات 'Problem Solving'، التي تهدف إلى تمكينك من أداء عملك بفاعلية بغض النظر عن التحديات والعقبات. هذه القدرة على حل المشكلات هي ما يتمتع به العاملون في 'جمال تك'، ومن الضروري أن تطور في نفسك هذه المهارة لتصبح عضوًا فعالًا في فريق عمل 'جمال تك'.

Gammal Tech's Network Optimizer

Background

Gammal Tech, a pinnacle in the software development industry, renowned for its innovative approach and state-of-the-art office facilities, is at the forefront of developing groundbreaking technologies. Their latest venture is into optimizing network systems to enhance communication efficiency within their sophisticated infrastructure.

Problem Statement

Gammal Tech's R&D team is working on a new technology, the "Network Optimizer", designed to reduce latency in their internal communication network. The Optimizer aims to reconfigure the network by identifying the optimal path between any two systems within the network, minimizing the data travel time.

Your task is to develop a program that takes the layout of Gammal Tech's network as input and determines the shortest path between two given systems in terms of latency. Each connection between systems has a latency value.



Input Format

- The first line contains two integers, N and M , where N is the number of systems (numbered 1 through N) and M is the number of connections.
- The next M lines each contain three integers, a , b , and w , indicating a connection between system a and system b with a latency of w milliseconds.
- The last line contains two integers, S and D , representing the source and destination systems.

Output Format

- Output a single line containing the minimum latency (in milliseconds) from system S to system D .

Constraints

- $1 \leq N \leq 1000$
- $1 \leq M \leq 10000$
- $1 \leq S, D \leq N$
- $1 \leq w \leq 1000$

Sample Input:

```
5 6
1 2 10
1 3 5
2 4 1
3 4 2
3 5 5
4 5 2
1 5
```

Sample Output:

```
7
```



Explanation

The shortest path from system 1 to system 5 is 1 -> 3 -> 4 -> 5 with a total latency of 7 milliseconds.

للتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمرين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق

C Programming Solution:

```
#include <stdio.h>
#include <limits.h>

#define MAX_N 1000

int dist[MAX_N];
int graph[MAX_N][MAX_N];
int visited[MAX_N];

int minDistance(int n) {
    int min = INT_MAX, min_index;
    for (int v = 0; v < n; v++) {
        if (!visited[v] && dist[v] <= min) {
            min = dist[v];
            min_index = v;
        }
    }
    return min_index;
}

void dijkstra(int src, int n) {
    for (int i = 0; i < n; i++) {
        dist[i] = INT_MAX;
        visited[i] = 0;
    }
    dist[src] = 0;

    for (int count = 0; count < n - 1; count++) {
        int u = minDistance(n);
        visited[u] = 1;
        for (int v = 0; v < n; v++) {
            if (!visited[v] && graph[u][v] && dist[u] != INT_MAX &&
                dist[u] + graph[u][v] < dist[v]) {
                dist[v] = dist[u] + graph[u][v];
            }
        }
    }
}
```



```
int main() {
    int n, m;
    scanf("%d %d", &n, &m);

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            graph[i][j] = 0;
        }
    }

    for (int i = 0; i < m; i++) {
        int a, b, w;
        scanf("%d %d %d", &a, &b, &w);
        a--; b--;
        graph[a][b] = w;
        graph[b][a] = w; // Assuming the network is undirected
    }

    int s, d;
    scanf("%d %d", &s, &d);
    s--; d--;

    dijkstra(s, n);

    printf("%d\n", dist[d]);
    return 0;
}
```