# A. Requirements

## Code (90%)

You can write your code in Java, Python, C, or C++. The *time limit* may vary among different languages, depending on the performance of the language. Your code must be a complete excutable program instead of only a function. We guarantee test data strictly compliance with the requirements in the description, and you do not need to deal with cases where the input data is invalid.

**No AI Assistance or Plagiarism:** All code must be your own. The use of AI tools (e.g., ChatGPT, GitHub Copilot) or copying from external sources or peers is **strictly forbidden**.

Violations of the plagiarism rules will result in 0 points or even **failure** of this course.

**Libraries in this assignment:**

- For C/C++, you can only include standard library.
- For Java, you can only `import java.util.*`
- For Python, you can only import standard library. In other words, you cannot import libraries such as `numpy`.

We provide an example problem to illustrate the information above better.

## Report (10%)

You also need to write a report in `pdf` type to explain the following:

- What are the possible solutions for the problem?
- How do you solve this problem?
- Why is your solution better than others?

Please note that the **maximum** number of pages allowed for your report is **5 pages**.

Remember that the report is to illustrate your thinking process. Keep in mind that your report is supposed to show your ideas and thinking process. We expect clear and precise textual descriptions in your report, and we do not recommend that you over-format your report.

# B. Example Problem: A + B Problem

## Description

Given 2 integers A and B, compute and print A + B

## Input

Two integers in one line: A, and B

## Output

One integer: A + B

### Sample Input 1

```
1 2
```

### Sample Output 1

```
3
```

### Problem Scale & Subtasks

For 100% of the test cases, $0 \leq A, B \leq 10^6$

### Solutions

#### Java

```java
import java.util.*;

public class Example {
    public static void main(String[] args) {
        int a, b;
        Scanner scanner = new Scanner(System.in);
        a = scanner.nextInt();
        b = scanner.nextInt();
        scanner.close();
        System.out.println(a + b);
    }
}
```

#### Python

```python
AB = input().split()
A, B = int(AB[0]), int(AB[1])
print(A + B)
```

#### C

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
  int A, B;
  scanf("%d%d", &A, &B);
  printf("%d\n", A + B);
  return 0;
}
```

#### C++

```cpp
#include <iostream>

int main(int argc, char *argv[])
{
  int A, B;
  std::cin>> A >> B;
  std::cout<< A + B << std::endl;
  return 0;
}
```

## C. Submission

After finishing this assignment, you are required to submit your code to the Online Judge System (OJ), and upload your `.zip` package of your code files and report to BlackBoard.

### C.1 Online Judge

Once you have completed one problem, you can submit your code on the page on the Online Judge platform (`oj.cuhk.edu.cn`, campus only) to gain marks for the code part. You can submit your solution of one problem for **no more than 80 times**.

After you have submitted your program, OJ will test your program on all test cases and give you a grade. The grade of your latest submission will be regarded as the final grade of the corresponding problem. Each problem is tested on multiple test cases of different difficulty. You will get a part of the score even if your algorithm is not the best.

**Note:** The program running time may vary on different machines. Please refer to the result of the online judge system. OJ will show the time and memory limits for different languages on the corresponding problem page.

If you have other questions about the online judge system, please refer to OJ wiki (campus network only). If this cannot help you, feel free to contact us.

### C.2 BlackBoard

You are required to upload your **source codes and report** to the BlackBoard platform. You need to name your files according to the following rules and compress them into `A4_<Student ID>.zip` :

```
A4_<Student ID>.zip
|-- A4_P1_<Student ID>.java/py/c/cpp
|-- A4_P2_<Student ID>.java/py/c/cpp
|-- A4_Report_<Student ID>.pdf
```

For Java users, **you don't need to consider the consistency of class name and file name**.

For example, suppose your ID is 123456789, and your problem 1 is written in `Python`, problem 2 is written in `Java` then the following contents should be included in your submitted `A4_123456789.zip`:

```
A4_123456789.zip
|-- A4_P1_123456789.py
|-- A4_P2_123456789.java
|-- A4_Report_123456789.pdf
```

### C.3 Late Submissions

Submissions after Dec.08 2024 23:59:00(UTC+8) would be considered as LATE.

The LATE submission page will open after deadline on OJ.

Submisson time = max{latest submisson time for every problem, BlackBoard submisson time}

There will be penalties for late submission:

- 0–24 hours after deadline: final score = your score×0.8
- 24–72 hours after deadline: final score = your score×0.5
- 72+ hours after deadline: final score = your score×0

## FAQs

**Q:** My program passes samples on my computer, but not get AC on OJ.

**A:** Refer to OJ Wiki Q&A

## Authors

If you have questions for the problems below, please contact:

- Problems 1-3. Xinyan Zhou: 123090908@link.cuhk.edu.cn
- Problems 1-3. Chunxu Lin: 221012033@link.cuhk.edu.cn

# CSC3100 Data Structures Fall 2024

**Programming Assignment 4**

Xinyan Zhou: 123090908@link.cuhk.edu.cn
Chunxu Lin: 221012033@link.cuhk.edu.cn

Due: Dec.08 2024 23:59:00

**Assignment Link:** `https://oj.cuhk.edu.cn/d/csc3100_2024_fall/homework/67432a4f92b47bee0be02705`

**Please note that:** you can choose **any two questions from the three questions** to get at most 90% of the score, and the remaining score is from the report.

## 1 Mirror

### 1.1 Description

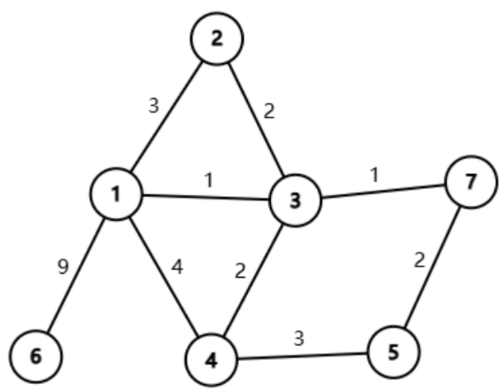



Figure 1: An example map of the city

Lee wants to show his new invention, a special mirror, to his friends. These friends are scattered over the city, which consists of $n$ nodes connected by $m$ **undirected edges**. It takes $w_i$ time to pass through an edge, $i = 1, ..., m$. Figure 1 shows a map of a city. For a path from the starting node $s$ to the ending node $t$, Lee decided that he must go through $k$ specific undirected edges $E = \{e_1, e_2, \ldots, e_k\}$. Please note that:

1. There is no pass order requirement for these specific edges;
2. If $e_1 = (u, v)$, Lee could pass the edge from $u$ to $v$, he also could pass the edge from $v$ to $u$.

Lee could start from different nodes and end at different nodes. In addition, a pair consisting of a starting node $s_j$ and an ending node $t_j$ corresponds to a set of edges $E_j$, $j = 1, ..., q$, which Lee must pass through. Since time is of the essence in the city, Lee needs to find the shortest possible path through all the specified edges.

Here is a detailed example around Figure 1 with $q = 2$:

1. Lee starts from node $s_1 = 1$ and ends at node $t_1 = 5$. Besides, he is required to pass through an undirected edge $E_1 = \{(1, 6)\}$(i.e. edge between 1 and 6). Then, his path is $1 \to 6 \to 1 \to 3 \to 7 \to 5$;
2. When Lee starts from node $s_2 = 2$ and ends at node $t_2 = 7$, he has to pass through two undirected edges $E_2 = \{(1, 3), (1, 2)\}$. Therefore his path is $2 \to 1 \to 3 \to 7$.

## 1.2 Input

- The first line contains three integers $n$, $m$, and $q$, indicating the number of nodes $n$, the number of edges $m$, and the number of planned paths $q$.
- The next $m$ lines each contain three integers $u$, $v$, and $w$. These represent an edge between nodes $u$ and $v$, with a traversal time of $w$. The edge described in the $i$-th line is numbered $i$.
- For each of the next $q$ blocks, the first line contains an integer $k_i$. The following line contains $k_i$ integers $e_1, e_2, \ldots, e_{k_i}$, indicating the indices of the edges Lee must pass through.
- The next $q$ lines each contain two integers $s_i$ and $t_i$, indicating the starting point $s_i$ and the ending point $t_i$ of each path.

## 1.3 Output

- $q$ lines, $i$-th line indicating the smallest amount of time of $i$-th paths.

### Sample Input 1

```
7 9 2
1 2 3
5 4 3
3 1 1
6 1 9
3 4 2
1 4 4
3 2 2
3 7 1
5 7 2
1
4
2
1 3
1 5
2 7
```

### Sample Output 1

```
22
5
```

The graph of Sample 1 is shown in Figure 1.

### Sample Input 2

```
See attached q1sample2.in
```

### Sample Output 2

```
See attached q1sample2.out
```

### Sample Input 3

```
See attached q1sample3.in
```

### Sample Output 3

```
See attached q1sample3.out
```

## Problem Scale & Subtasks

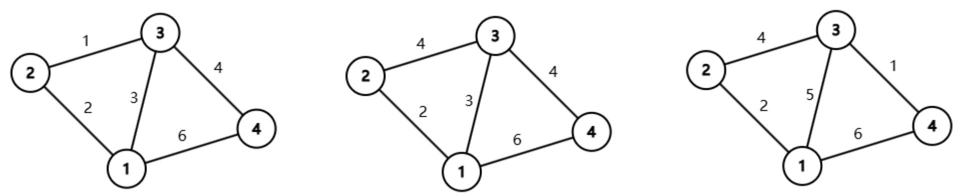
- $k \leq 5$, $q \leq 100$, $w_i \leq 2 \times 10^3$,

- $n \leq 1000$, $n-1 \leq m \leq 2 \times 10^3$, and $1 \leq s, t \leq n$

| Test Case No. | Constraints |
|---|---|
| 1-2 | $m, n \leq 100$ |
| 3-5 | $n \leq 500$ |
| 6-10 | $n \leq 1000$ |

# 2 Violet

## 2.1 Description




(a) The graph of 0-th day (b) The graph of 1-th day (c) The graph of 2-th day

Figure 2: An example map of the park.

Heath wants to have a date with his beloved Kathy. He decides to invite Kathy for a walk in a huge park. This park can be described as an **undirected dynamic graph**, consisting of $n$ flowerbeds and $m$ undirected edges connecting these flowerbeds. Each edge is filled with violets, Kathy's favorite flower. The $i$-th edge has $w_i$ violets, $i = 1, ..., m$.

There are $q+1$ days Heath could choose to have the date. Besides, at the beginning of $i$-th day, $i = 0, 1, 2, ..., q$, the number of violets on a set of edges will **change**, where $k_i$ edges will be affected. Please note that $i = 0$ indicating the park has **not changed** yet.

Heath wants to plan a no-duplicate-node path where he and Kathy will start from point $s_i$ and end their tour at point $t_i$. He hopes to maximize the minimum number of violets encountered on any no-duplicate-node path throughout the entire path. You are expected to give him the maximum number of these values each day.

A detailed example around Figure 2 with $q = 2$ is provided in the following. Heath will choose one of the 3 days to date:

- At the beginning of 0-st day, the graph is shown in Figure 2(a);
- At the beginning of 1-st day, $k_1 = 1$. The value on edge $(2, 3)$ changes to 4;
- At the beginning of 2-nd day, $k_2 = 2$. The value on edge $(3, 4)$ changes to 1 and the value on edge $(1, 3)$ changes to 5;

Take Figure 2(a) as an example, Heath wants to start from 1 and end at 4 on 0-th day. There are three no-duplicate-node paths he can choose to date:

1. $1 \to 2 \to 3 \to 4$, with the number of violets on the path being 2, 1, and 4 respectively, and the smallest number is 1;
2. $1 \to 3 \to 4$, with the number of violets on the path being 3, 4 respectively, and the smallest number is 3;
3. $1 \to 4$, with the number of violets on the path being 6, and the smallest number is 6.

Therefore, we choose the path with the largest minimum number, which is $1 \to 4$, and the answer is 6.

## 2.2 Input

- The first line contains two integers $n$ and $m$, indicating there are $n$ flowerbeds and $m$ paths;
- The next $m$ lines each contain three integers $u$, $v$, and $w$, representing a path between flowerbeds $u$ and $v$, lined with $w$ violets;
- The next line contains an integer $q$, the number of changes that will occur;
- For each of the next $q$ blocks, the first line contains an integer $k_i$, the number of paths affected by the change. The following $k_i$ lines for each contain three integers $a$, $b$, and $c$, indicating that the path between flowerbeds $a$ and $b$ now has $c$ violets;
- For next $\mathbf{q+1}$ lines, there including two integer $s_i$, $t_i$ indicating that Heath will start from flowerbed $s_i$ and end at flowerbed $t_i$.

## 2.3 Output

- $q+1$ integer, indicating the maximum number of fewest violets after every change.

### Sample Input 1

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

### Sample Output 1

```
6
4
4
```

The dynamic graph of Sample 1 is shown in Figure 2.

### Sample Input 2

```
See attached q2sample2.in
```

### Sample Output 2

```
See attached q2sample2.out
```

### Sample Input 3

```
See attached q2sample3.in
```

### Sample Output 3

```
See attached q2sample3.out
```

### Problem Scale & Subtasks

- $1 \le s_i, t_i \le n$,
- $m \le 2 \times 10^4$, $q \le 100$, $w_i \le 10^4$, and
- Heath can always reach $t$ starting from $s$.

| Test Case No. | Constraints |
|---|---|
| 1-2 | $m, n \le 100$ and $k_i = 1$ |
| 3-5 | $n \le 100$ and $k_i \le 10$ |
| 6-10 | $n \le 10^4$ and $k_i \le 10$ |

# 3 Hero

## 3.1 Description

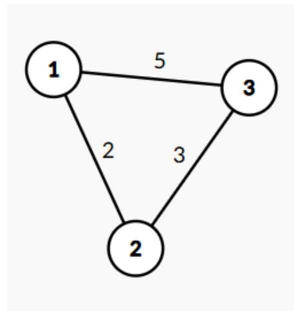



Figure 3: An example map of La Mancha Land

Don plays the role of a hero in the La Mancha Land parade, defeating monsters and obtaining treasures. Don has two attributes: health points (HPs) and spirit points (SPs). Since Don is very powerful, she can decide these values herself. Her adventure map can be viewed as an undirected simple connected graph with $n$ nodes and $m$ edges. Don needs to start from her hometown $s$ and reach the treasure location $t$.

Each time she passes an edge, her spirit point decreases by 1. Additionally, each edge has a monster with an attack power of $a_i$. If Don's spirit point is $t$, she loses $\lfloor \frac{a_i}{t} \rfloor$ health points. The monsters cannot be defeated. Therefore, if Don passes through the same edge twice, she will suffer damage twice.

To entertain the parade-goers, she wants to reach the destination at the most desperate moment, which means her **spirit and health points are both exactly 0**. While preparing her costume for the performance, she asks you to determine the **minimum initial health points** she needs to achieve her goal.

Taking Figure 3 as an example, if Don starts from node 1 and ends at node 3, she has two paths to choose from:

- $1 \to 2 \to 3$: at node 1, Don has SP = 2 and HP = 4; and at node 2, Don has SP = 1 and HP = 3; and at node 3, his SP = 0 and HP = 0, which satisfies the requirement.
- $1 \to 3$: at node 1, SP = 1, HP = 5; at node 3, SP = 0, HP = 0.

Therefore, the minimum number of health points Don needs to maintain at the starting node is 4.

## 3.2 Input

- The first line contains four integers $n$, $m$, $s$, and $t$, indicating the number of nodes $n$, the number of edges $m$, Don starts from node $s$ and ends at node $t$;
- The next $m$ lines each contain three integers $u$, $v$, and $a_i$, representing an edge between node $u$ and $v$, the attack power of monster is $a_i$.

## 3.3 Output

- One integer, indicating the least health point Don needs.

**Sample Input 1**

```
3 3 1 3
1 2 2
1 3 5
3 2 3
```

**Sample Output 1**

```
4
```

The graph of Sample 1 is shown in Figure 3.

### Sample Input 2

```
See attached q3sample2.in
```

### Sample Output 2

```
See attached q3sample2.out
```

### Sample Input 3

```
See attached q3sample3.in
```

### Sample Output 3

```
See attached q3sample3.out
```

## Problem Scale & Subtasks

| Test Case No. | Constraints |
|---|---|
| 1-2 | $m, n, \alpha_i \leq 10$ |
| 3-6 | $n \leq 10^3$, $m \leq 2 \times 10^3$ and $\alpha_i \leq 10$ |
| 7-8 | $n \leq 2 \times 10^4$, $m \leq 4 \times 10^4$, and $\alpha_i = 1$ |
| 9-10 | $n \leq 2 \times 10^4$, $m \leq 4 \times 10^4$, and $\alpha_i \leq 100$ |