

## Programming Assignment #4

# The Map Maker

---

Due: 23:59 Sunday, December 29

## Problem Description

You are an apprentice in map planning. To reduce the workload of your master, you decide to automate the planning process using your algorithm knowledge and coding skills.

After organizing the maps, you find that the maps are from four different locations and have different goals. The locations and goals are listed below:

- **Maps of Los Santos**  
The maps consist of  $|V|$  buildings and  $|E|$  transportation routes. Each route  $E_k$  represents a method to travel from building  $V_i$  to  $V_j$  in exactly  $t_k$  minutes. Given a safe house  $V_0$ , Trevor, a successful dealer, demands you to calculate how fast one can travel to  $V_0$  starting from each building.
- **Maps of Portal Chambers**  
The maps consist of  $|V|$  chambers and  $|E|$  teleportation routes. One can travel to other chambers by teleporting through portals. Each teleportation route  $E_k$  travels from chamber  $V_i$  to  $V_j$  with time difference  $t_k$ . Some  $t_k$  may be negative numbers indicating traveling back in time due to the side effect of the portal. However, there isn't a sequence of teleportation routes that allows one to keep going back in time. Chell, a young researcher, needs your help to calculate the minimum amount of time to travel from any chamber  $V_i$  back to that same chamber through multiple teleportations.
- **Maps of Skyrim**  
The maps consist of  $|V|$  inns and  $|E|$  transportation paths. Each path  $E_k$  represents a way to travel from inn  $V_i$  to  $V_j$  that uses up  $f_k$  pounds of meat. However, there are a variety of animals when traveling through the routes, we can gain  $g_k$  pounds of meat by hunting animals when traveling through each route  $E_k$ .  $t_k$  is defined to be  $g_k - f_k$  for ease of calculation. Aela, a skillful huntress, wants you to calculate whether the map contains a sequence of routes that can keep accumulating meat by hunting animals, assuming that one

can always acquire the same amount of meat even when walking through a same path the second time.

- **Maps of Blight Town (Bonus)**

The maps consist of  $|V|$  drylands and  $|E|$  routes. Each route  $E_k$  represents a way to travel from dryland  $V_i$  to  $V_j$ , while  $t_k$  indicates whether the route requires one to walk through the poisonous swamp. Since different warriors have different poison resistance and each map has a different kind of poison, the condition for the warrior to be poisoned varies. If there are  $m + 1$  routes in the  $n$  recent chosen routes that require walking through the poisonous swamp, the warrior will be poisoned and requires special treatment. Solaire, a kind warrior, asks you to calculate whether there exists a route for one to travel from  $V_0$  to  $V_{|V|-1}$  without getting poisoned.

Hints:

- Being able to travel from  $V_i$  to  $V_j$  doesn't mean one can travel back ( $V_j$  to  $V_i$ ) using the same route. For example, there may be one-way busses in Los Santos and border controls in Skyrim.
- The request of the first three maps (Los Santos, Portal Chambers, Skyrim) can be solved through basic algorithms learned in class. The last map (Blight Town) can be optimized through various techniques.

## **I/O Format**

Use standard I/O. (stdin, stdout)

### **Input**

The first line contains one integer  $T$  ( $1 \leq T \leq 10$ ), representing the number of test cases  $T$ . Then  $T$  test cases follow.

For the first line of each test case, there are 3 positive integers  $K, V, E$ .  $K$  represents the map kind, 1 for Los Santos, 2 for Portal Chambers, 3 for Skyrim, and 4 for Blight Town.  $V$  and  $E$  is the  $|V|$  and  $|E|$  in the problem description.

If  $K = 4$ , there are an additional line with 2 integers  $n, m$ .

Then there are  $E$  lines, the  $k^{th}$  line have 3 integers  $i, j, t_k$ , representing the route  $V_i$  to  $V_j$  with parameter  $t_k$ . The meaning of  $t_k$  depends on the problem description.

Your code will be graded by multiple test cases:

- 30%:  $K = 1, 2 \leq |V| \leq 10^5, 0 \leq |E| \leq 10^6, 0 \leq t_k \leq 10^9$ .
- 30%:  $K = 2, 2 \leq |V| \leq 10^3, 0 \leq |E| \leq 10^6, -10^9 \leq t_k \leq 10^9$ .
- 30%:  $K = 3, 2 \leq |V| \leq 10^4, 0 \leq |E| \leq 10^5, -10^9 \leq t_k \leq 10^9$ .
- Bonus 40%:  $K = 4, 2 \leq |V| \leq 10^9, 0 \leq |E| \leq 2 \cdot 10^9, 0 \leq t_k \leq 1, 0 \leq m \leq n \leq 30$ .

The score depends on the time and how many test cases you can pass; optimizations are encouraged.

## Output

For each test case  $T$ , output the required result:

- For Maps of Los Santos ( $K = 1$ )  
Output the shortest path for  $V_i$  to  $V_0$  for  $i = 0, 1, \dots, |V| - 1$ . (output INF if there are no path from  $V_i$  to  $V_0$ )
- For Maps of Portal Chambers ( $K = 2$ )  
Output the shortest loop for  $V_i$  to  $V_i$  for  $i = 0, 1, \dots, |V| - 1$ . (output INF if there are no loop from  $V_i$  to  $V_i$ )
- For Maps of Skyrim ( $K = 3$ )  
Output whether there is a special route sequence. (output TRUE or FALSE)
- For Maps of Blight Town ( $K = 4$ )  
Output whether there is a route from  $V_0$  to  $V_{|V|-1}$  without getting poisoned. (output TRUE or FALSE)

## Examples

### Input 1

```
1
1 5 5
0 2 10
1 0 1
3 0 3
3 1 1
4 0 2
```

### Output 1

```
0
1
INF
2
2
```

**Input 2**

1  
2 3 6  
0 1 2  
0 2 5  
1 0 8  
1 2 10  
2 0 2  
2 1 -8

**Output 2**

5  
2  
2

**Input 3**

1  
3 3 3  
0 1 -4  
1 2 -2  
2 0 7

**Output 3**

TRUE

**Input 4**

1  
4 5 6  
3 2  
0 1 1  
1 2 0  
1 3 1  
2 1 0  
2 3 1  
3 4 1

**Output 4**

TRUE

## Program Submission

1. Please use C/C++ and write your program in **a single source file**, you may use the given template: "hw4\_template.cpp".
2. Your source file must be named as "**<Student\_ID>\_hw4.cpp**" and please make sure that all characters of the filename are in **lower case**. For example, if your student id is 106062000, the name of your program file should be 106062000\_hw4.cpp.
3. Your program will be compiled in a GNU/Linux environment with:  
`g++ -O2 -std=c++14 <Student_ID>_hw4.cpp`
4. The source file must be uploaded directly, without compressing the file.
5. **0 points will be given to Plagiarism. NEVER SHOW YOUR CODE** to others and you must write your code by yourself. If the codes are similar to other people and you can't explain your code properly, you will be identified as plagiarism.

## Report

1. Your report must contain the flowchart or the pseudo code of your program. You have to describe how your approach works. The time complexity analysis of the implemented algorithm should be included as well.
2. The report filename must be "**<Student\_ID>\_hw4.pdf**" and please make sure that all characters of the filename are in lower case.

## Grading Policy

You must submit both your source code and report. Remember the submission rules mentioned above, or you will be punished on your grade.

● Test cases	90%
● Report	20%
● Bonus	40%