CP for AP

CS 293

Autumn 2022

Contents

1 Problem 1 - Useless Cities

Problem Statement

Aegon needs to find his sister Rhaenyra in order to eliminate all challenges to his claim on the throne.

Westeros could be viewed as a country with N cities. These cities are joined by M segments of the Kingsroad that Aegon would be following. Aegon initally rests in King's Landing which is the 0^{th} Node, whereas Rhaenyra is at Castle Black which is the $(N-1)^{th}$ Node.

The King(Aegon) would move only along the Kingsroad, and would move along the shortest path.

Being the Barbarian that Aegon is, He wants to watch some cities burn. Therefore, He demands of you(his hand!) the indices of cities which are *useless*.

A city (represented by a Node in the said Graph) is said to be *useless* if we can remove this city from the graph(and Westeros!) and the length of the shortest path from King's Landing to Castle Black is unchanged.

Input Format

The First Line would contain two integers N and M, The Number of Cities in Westeros, and the number of segments of the Kingsroad which connect them.
The Next M lines would contain three integers u, v, l which means that a segment of the Kingsroad can take you from Node u to Node v, which has length l.

Note that these segments of the Kingsroad are uni-directional.

Output Format

On a single line output the Number of such cities which can be removed without affecting the length of the shortest path from Node 0 to Node $N-1$. In the next line print the indices of these cities separated by spaces. These indices should be sorted in increasing order.

Constraints

$1 \leq N \leq 10^5$,
$1 \leq M \leq 2 \times 10^5$,
$0 \leq u, v < N$
$1 \leq l \leq 10^9$

Sample Input

```
7 7
0 1 10
1 2 8
2 3 12
1 4 9
4 5 11
5 6 10
3 6 10
```

Sample Output

```
4
2 3 4 5
```

Explanation

You may remove any of 2,3,4 or 5, and the length of the shortest path from Node 0 to Node $N-1$ in rest of the Graph won't change.

2 Problem 2 - Last One Standing

Problem Statement

Azkaban has N prisoners, standing in a circle, up for execution. The Executioner is in a good mood today and will spare the prisoner who is the last.

The order of execution is random, but it would be reported to you in a said manner (more on this in the *Input Format* section!).

Your task is simple, output the name of the last prisoner standing.

Input Format

The First Line would contains an integer N, the number of prisoners in Azkaban.
The next N lines enlist the names of the N prisoners.
The next $N-1$ lines would contain integers $a_i : i = 1, 2, \ldots (N-1)$

a_1 specifies the index of the first prisoner to be exectued. (the index is 0-based)

$a_i : i = 2, 3 \ldots (N-1)$ specifies the index of the prisoner to be executed relative to the prisoner that was executed last. This index is with reference to the new circular queue, after removing the prisoners that are already executed. Eg. If my queue looked like "ABC" and "B" was executed, and the next index is 1, then "C" would be executed next. Furthermore, this number given to you can be large, and in that case we expect you to identify the value of the next index to be executed modulo the current size of the circular-queue.

Output Format

In a single line, output the name of the last prisoner standing.

Contraints

$1 \leq N \leq 10^5$,
$a_1 \geq 0$,
$1 \leq a_i \leq 10^9$

Sample Input

6
Bellatrix
Sirius
Peter
Igor
Barty
Lucius
2
14
3
89
45

Sample Output

Lucius

Explanation

The order of execution is : Peter $\rightarrow$ Bellatrix $\rightarrow$ Barty $\rightarrow$ Sirius $\rightarrow$ Igor

3 Problem 3 - Teleporter Transport

Problem Statement

You have enjoyed an entire semester of rail travel planning :). Here, we consider a transportation network problem comprising teleporters and roads.

Consider a kingdom which has n towns lying in a 2-dimensional plane. The king wants to construct a network of roads and teleporters such that it is possible for people to travel from any city to any other city. He assigns you, his minister, to calculate the minimum cost required to do so. Can you compute this least cost?

Each road acts as a bidirectional mode of travel implying if there is a road between city i and city j, then it is possible to travel from city i to city j and vice-versa. Each teleporter allows a person to travel from any city having a teleporter to any other city which **also** has a teleporter. Each city is associated with a 3-tuple (x_i, y_i, p_i) where x_i, y_i represent the x and y coordinates of city i respectively. p_i represents the cost of constructing a teleporter in city i. The cost of constructing a direct road between cities i and j, c_{ij} is given by

$$c_{ij} = \alpha(|x_i - x_j| + |y_i - y_j|)$$

where α is a given input parameter. Note that a pair of cities need not have a direct connection (either a direct road or teleporter). They can be connected indirectly via many intermediate cities, say through a network of roads and teleporters.

Input Format

The first line contains an integer T containing number of testcases. **Within** each testcase, the first line contains 2 integers representing the number of cities n and the parameter α. The next n lines contain 3 integers each representing x_i, y_i, p_i for the n cities respectively. The next testcase begins on the next line. Input from standard input. You can check the sample input files provided.

Output Format

For each testcase, output a single integer on a new line corresponding to the minimal cost in each testcase. Output to standard output. You can check the sample output files provided

Constraints

$1 \leq T \leq 50$
$\sum_T n^2 \leq 10^7$
$0 \leq p_i \leq 10^4$
$0 \leq \alpha \leq 100$
$-10^5 \leq x, y \leq 10^5$

Example

```
Input:
1
4 1
100 0 5
200 0 10
300 0 250
400 0 20
```

```
Output:
135
```

Explanation

We construct a teleporter at A, B and D and a road between B and C or between C and D to get a minimal cost as 135.

4 Problem 4 - Identify the First K

Problem Statement

Frustated by a hectic-semester, You head to your FacAD saying you can't take it anymore. Your claim is that you deserve an AA in every course as you already know the content too well, and you don't want to waste time solving assignments anymore.

Your FacAD gives you the following problem to test your calibre:

He has an array A of N distinct real numbers in mind. You will be given N but not A. You are to ask him a series of queries and eventually reveal the indices of the largest K (K being even) elements in A using $K/2$ rounds of interaction. A single round of interaction comprises of a sequence of queries you can ask your facad and the round concludes by you revealing two new (not previously revealed) indices to your facad. Let the rounds of interaction be numbered from 0 through $K/2-1$. In the i^{th} round, you can ask your facad one or more queries of the form `Query(l, r)`, where $0 \leq l \leq r < N$ and $(r-l+1) \geq K-2i$, and your facad will return you the index of the $(K-2i)^{th}$ largest element in $A[l]$ through $A[r]$. You can ask as many queries as you want in a round as long as the total number of queries you ask across all $K/2$ rounds doesn't exceed $K * \lceil \log_2 N + 5 \rceil$. A round ends when you reveal 2 distinct not previously revealed indices, say index1 and index2, among the indices of the largest K elements in A by invoking the function `Reveal(index1, index2)` (the function would return `True`, if you are correct or terminate otherwise).

A typical interaction looks like this :

Query(a,b) *Round 0 starts: Returns index of K^{th} largest integer in index range $[a, b]$*
Query(c,d) *Returns index of K^{th} largest integer in index range $[c, d]$*
⋮
Reveal(e,f) *Round 0 ends: You know for sure that the indices e and f are among the top K*
Query(g,h) *Round 1 starts:Returns index of $(K-2)^{th}$ largest integer in index range $[g, h]$*
⋮
Reveal(i,j) *Round 1 ends: You know for sure that the indices i and j are among the top K*
⋮
Reveal(p,q) *Round $K/2-1$ ends: You know for sure that the indices p and q are among the top K*

Input Format

You need to implement the function `giveMetheTopK(int N, int K)`, which would be provided N and K as input and can make calls to `Query` and `Reveal`.

Constraints

$1 \leq N \leq 10^6$,
$1 \leq K \leq \sqrt{N}$
1 ≤Number of Queries $\leq K * \lceil \log(N) + 5 \rceil$
Suppose you make queries for $Q = \{(i_1, j_1), (i_2, j_2) \ldots, (i_q, j_q)\}$ then $\sum_Q (j - i + 1) \leq 10^8$ (Do not worry alot about this!)

Example

5 2
$Query(0, 4) \rightarrow 1$
$Query(1, 4) \rightarrow 1$
$Query(2, 4) \rightarrow 2$
$Query(1, 3) \rightarrow 1$
$Query(1, 2) \rightarrow 2$
$Reveal(1, 3) \rightarrow True$

Explanation

The Array is such that :
$A[0] = 1$
$A[1] = 4$
$A[2] = 3$
$A[3] = 5$
$A[4] = 2$

You can see that based on the Result of said Queries, We may easily prove that the answer we've given is correct!

5 Submission and Autograding

Problem 1

All your code to this problem must be in one file named "ROLLNO_uselessCities.cpp". Please follow this convention strictly. Upload this one single file under the appropriate assignment on moodle.

In order to check your submission, make sure that it is in the same directory as provided to you, and compile it to generate `exec` by running `g++ ROLLNO\_uselessCities.cpp -o exec`. You may then simply run :

```
time bash student_autograder.sh
```

Problem 2

All your code to this problem must be in one file named "ROLLNO_lastOneStanding.cpp". Please follow this convention strictly. Upload this one single file under the appropriate assignment on moodle.

In order to check your submission, make sure that it is in the same directory as provided to you, and compile it to generate `exec` by running `g++ ROLLNO\_lastOneStanding.cpp -o exec`.

You may then simply run :

```
time bash student_autograder.sh
```

Problem 3

All your code to this problem must be in one file named "ROLLNO_teleporters.cpp". Please follow this convention strictly. Upload this one single file under the appropriate assignment on moodle.

To run the autograder script we have provided, please run

```
g++ ROLLNO_teleporters.cpp -o exec
time bash student_autograder.sh
```

Note that this will only check for correctness. For checking time constraints, please read the section on benchmarking

Problem 4

All your code to this problem must be in one file named "ROLLNO_firstK.cpp". Please follow this convention strictly. Upload this one single file under the appropriate assignment on moodle.

In order to check your submission. You may change the values in `main`(currently set to 1000,10), compile using `g++ ROLLNO\_firstK.cpp` and run using `./a.out`

6 Benchmarking

Your device is most probably different from the one we will be testing this code on. You can run time.cpp to compare processor speeds and change the timeout condition appropriately. Note that timeout will roughly be set to 2 times the time it takes for time.cpp to run

```
g++ time.cpp -o timer
time ./timer
```