Big Homework 2
Data Structures and Algorithms Graphs and Trees

General mentions

For this project you will work either in teams of 2 people or alone.

The homework will be uploaded on the Moodle platform (curs.pub.ro), for Hw2 assignment.

If you encounter problems with the platform, contact the lab assistant responsible for that problem.

The homework must be submitted by 27.05.2020, at 23:59. No late submissions will be accepted.

You will be asked details about your solutions during the following lab.

All input data must be read from files, and the output data must be written to a file. (example : input.in and output.out)

The final submission will contain an archive named Student1FamilyName_Student1Name_ Student2FamilyName_Student2Name_HW2 with:

- **the source files of your project (.cpp and .h), grouped in separate folders for each exercise (and not contain the object files (.o) or executables (.exe) or codeblocks project files (.cbp))**
- **a README file in which you will specify all the functional sections of the project, together with instructions for the user; additionally, if you have parts of the homework that don't work, you may offer solution ideas for a partial score on these sections.**

Warning: we will use plagiarism detection software on your submissions (Stanford's tool Moss). Copied homework will be marked with 0 points.

Observation: The classes used will be in headers (.h) and will be generic classes (template).

EX1 : Word chain game

Ana and Bogdan invented a new Word chain game. They choose a word list and try to form a chain with all the words in the list so that any two consecutive words in the chain follow the Word chain rule (the last letter of the first word coincides with the first letter of the second word).

Requirement

Check if a chain can be built for a given word list that meets the conditions in the statement.

Input data

The `input.in file` contains the natural number `T` on the first line , representing the number of tests in the file. The following is a description of the `T` tests. For each test, the natural number `N` representing the number of words in the list is written on a line , then the `N` words in the list, one word on a line.

Output data

The `output.out file` will contain `T` lines, one for each test in the input file. Line `i` contains the result for the `i-` th test in the input file: the value `1` , if it is possible to construct the word chain according to the conditions in the statement, respectively the value `0` otherwise.

Limitation

- `1 ≤ T ≤ 10`
- `1 ≤ n ≤ 100000`
- A word consists of at least two and at most 1000 lowercase letters of the English alphabet.
- The same word may appear more than once in the word list.

Examples

input.in	**output.out**
3 4 ana bogdan nicoleta adam 2 tu eu 2 oo oo	1 0 1

If you have any questions, please e-mail: dianascurtu14@gmail.com

EX2 : Space Shuttle

The space shuttle USS Enterprise has to complete a tour between N planets, numbered from 1 to n, planet 1 being the start point, the Earth. The spaceship is transporting a very precious cargo, which cannot spend too much time in zero gravity, and has to be transported to the destination as soon as possible. At each planet, the ship has to wait for a specific amount of time, until all the cargo is loaded on the ship. In order to express distances between the planets in space, we define the Astronomical Unit (AU) : 1 AU = average distance from Earth to Sun. The ship is equipped with very advanced engines, and has a known speed of 1 AU / hour.

The ship is equipped with K units of solid afterburner fuel. If going from planet i to planet j takes x hours, then using t units of afterburner can decrease the time taken to max(x-t, 0) seconds where max(a,b) denotes the greater of the two values between a & b. The afterburner can be used all at once or in multiples of 1 unit (you cannot use 0.5 units of afterburner fuel, since it is a solid fuel, not a liquid one)

If the ship respects this timeline, and uses the afterburners perfectly timed, what is the minimum sum of travelling time for all the cargo ? The travelling time equals the time it arrived at the destination minus the time it boarded the ship.

Please remember that the ship must take all the cargo to its destination.

Input Format

a file called input.in :

The first line contains 3 space separated integers n, m and K which indicate the number of planets, total number of cargo that the ship needs to load and the total units of afterburner fuel present aboard the Enterprise.

The second line contains n-1 space separated integers where the i-th integer indicates the distance between planet (i-1) to planet i.

m lines follow each containing 3 space separated integers. The ith line contains ti, pi and ei in that order indicating the arrival time of the cargo at pi at time ti with his destination being ei.

```
n m K
d1 d2 ... dn-1   // di: the distance between planet_i to planet_(i+1).
t1 p1 e1         // cargo 1 arrives at his boarding point at p1 and his destination is e1
t2 p2 e2        // cargo 2 arrives at his boarding point at p2 and his destination is e2
...
tm pm em
```

Constraints

0 < n <= 100000
0 < m <= 100000
0 <= K <= 10000000
0 < di <= 100
0 <= ti <= 10000000
1 <= si < ei <= n

Output Format

The minimal total travel time, written in the output.out file

Sample Input

3 3 2
1 4
1 1 3
2 1 2
5 2 3

Sample Output

9

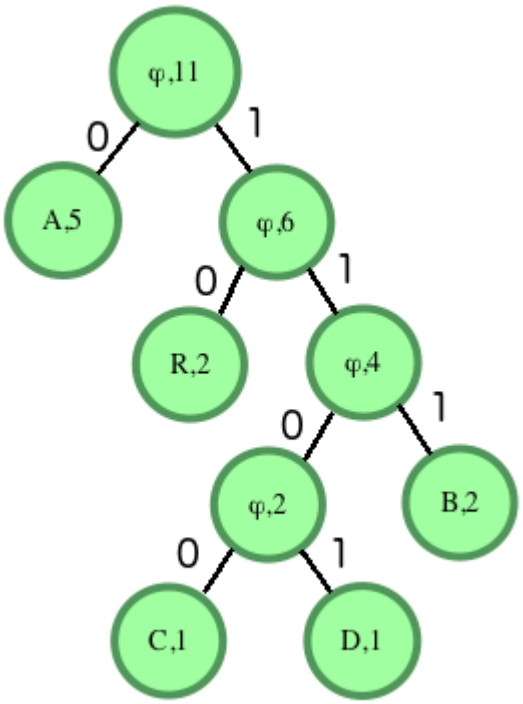
If you have any questions, please e-mail: alexandruhang@yahoo.com

Ex3: Decoder

Delta Force wants to create an encrypted message for Jason Bourne which Meduse will not be able to decode. Help them create the perfect algorithm for it!

They came with a crypto system which uses a tree to encrypt text. This system assigns variable length bits of output to fixed length input chars based on their frequency. For more frequent chars, a shorter output of bits will be created. All edges between the root and a char in the tree contain a code digit. If they are on the left side of the tree, the edge will have a 0 value and if they are on the right, the edge will have a 1 value. The chars will be located only on the leaves.

An example is the following one. Let's say that we want to encrypt the message: "ABRACADABRA". There are 11 characters. The frequency of each letter is the following one: A = 5, B = 2, R = 2, C = 1, D = 1. C and D are the smallest frequencies so create a tree with them. Their root will be the sum of their frequency, 1 + 1 = 2. The left node will always be the smaller value, the first alphabetical letter between the two, a tree. If you see a double frequency, use the first the letter appearing in the frequency computation (for example B will be first, then R)



Input: ABRACADABRA

Output:
A - 0
B - 111
C - 1100
D - 1101
R - 10

A B R A C A D A B R A
0 111 10 0 1100 0 1101 0 111 10 0

Your task is to do the encoding of such a message.

If you have any questions, please e-mail: oanca.a.m@gmail.com