Set phase 1

1. You are given N distinct coins. In one move you pick a random coin of value X and then

you remove all the coins with value X-1 and X+1 along with the coin X. The game gets

finished when no coins are left to pick. Find the maximum number of moves in which you

can finish the game. Constraints : 1 <= N <= 10^5

2. You are given a string S composed of only lowercase letters and an array of points

corresponding to each letter. You have to find maximum score you can get for a

substring which starts and ends with same letter. Constraints : 1 <= |S| <= 10^5 -10 <=

points <= 10

3. Implement lfu cache.

4. Given a board and ladders on the board at certain positions and a dice with 6

faces. You can climb a ladder if you are at starting position of ladder or if the

starting of a ladder is in between the start and end of another ladder.

It was asked to give the minimum number of rolls required to reach the end of the

board.

5. A boat when has full fuel tank can cover d stops. There are n stops and it’s given

in an array that which of the stops have fuel station. Boat can be stopped at such

stops to refill the tank. You have to minimize the number of stops to start from 1st

station and reach the nth station. I was able to submit this question, passing all

the test cases.

6. An inverted apple tree with two branches arising from each node is given.

Basically a binary tree. An apple can be normal, nearly overhydrated or neary

underhydrated. Penalties for overhydrated and underhydrated apple are given.

You have to water just one node in tree such that the sum of overhydrated and

underhydrated penalties for the whole tree is minimized. Input for tree is given in

the form of a parent array and another array denotes the status of each of the

apple node as 0 (neutral), -1 (under hydrated) and 1 (overhydrated). My naive

recursive code (without tree building) could pass only 5 out of 15 test cases.

7. There is a river of N unit and there are K stones across the river, each stone will

be in one of the N units.Starting from 0th location your first jump will always be 1

unit (consider previous jump as L units), now your next subsequent jump can be

either L + 1 or L – 1.Note few of the stoes can be missing.

You have to output the sum of square of Jump which you’ll take from each stone.

(The example will make it clear)

eg.let the river be of N = 5 units and there are 4 stones placed shown in array

[0, 1, 3, 4]

Solution –

Set phase 2

1.The first jump from 0-1 is always fixed which will be 1 Unit.

2.Now for the next jump you have 2 options L+1 = 2 or L-1 = 0, we have to move

forward from each location so it is not what we want, so we’ll check if current

location 1 + jump(L+1) i.e 1+2 = 3 is there in array or not, if yes we take a jump

3.Now from 3 we have 2 options (as last jump was 2 so we can jump 2+1 or 2-1

to reach next stone)

2-1 = 1 will make us reach to last stone 4 which we want

Now we have to output the sum of square of Jump’s taken at each location

0^2 + 1^2 + 2^2 + 1^2 = 6

So if it is possible to reach the last stone output 6 else output -1

The actual solution is using DP but I tried Greedy way(couldn’t pass all the

testcases)

8. You have been given a m\*n chess board and there were black pawns on the

board(represented by 1).You have to put your white pawn on all possible position’s

where it can be and the calculate how many kills are possible from each location and

sum it up.You can only move straight (x+1, y) and for a kill diagonal move (x+1, y+1) &

(x+1, y-1)

eg 3\*3

0 0 0

0 1 0

1 1 1

-First row from 1st position (0, 0) => 2 kills, 2nd position (0, 1) => 0 kills, 3rd position (0,

2) => 2 kills

-Second row, from 1st position (1, 0) => 1 kill, 2nd position (1, 1) => Not possible, 3rd

position (1, 2)=> 1 kill

-Third row not possible to place as all are 1

At each step, you have to maximize the kills

Output for the above question is – 6

-I had no time to solve this, even did not read the question properly

9. A and B play a game. They are given an array of positive numbers. Each player in his

turn picks up 2 numbers from the array such that the difference of the numbers does not

exist in the array. He then places the difference into the array too thus increase the

array count by 1. Then the next player repeats the same process. The game continues

till there are no 2 numbers such that difference does not exist in the array. The one

who’s not able to choose numbers loses. If A starts the game and the game proceeds

optimally, find who’ll win the game

Example : Input array : 2,5,3

A: 2,5,3,1

B: 2,5,3,1,4

A has no choice so B wins.

10. Given a string containing only lowercase alphabets, you have to convert it into a

string such that it contains only vowels by doing minimum number of operations. In one

operation, you could select a substring always starting from index 0, and move that

substring forward or backward. Example of rolling forward or backward are given :

Rolling Forward

Input- axzf

Let index chosen be 0 to 3 and moving it forward

Output- byag

Rolling Backward

Input – axze

Let index chosen be 0 to 2 and moving it backwards

Output- zwyd

11. (a) To find minimum number of operations to form the maximum length palindrome

string.

[28/02, 11:53 am] +91 98724 03283: Dear students the doc consists of two sets of coding problems.

These problems are very important from placement point of view so please complete them this weekend and submit your solutions.

Doubts are welcome

Dr. Rasmeet and Jagandeep Singh Sidhu

[28/02, 11:54 am] +91 98724 03283: Service now recent challenge problem no 1:

Read carefully and code submissions and doubts are awaited

You are given flights route map of a country consisting of N cities and M undirected flight routes. Each city has an airport and each airport can work as layover. The airport will be in two states, Loading and Running. In loading state, luggage is loaded into the planes. In the running state, planes will leave the airport for the next city. All the airports will switch their states from Loading to Running and vice versa after every T minutes. You can cross a city if its airport state is running. Initially, all the airports are in running state. At an airport, if its state is loading, you have to wait for it to switch its state to running. The time taken to travel through any flight route is C minutes. Find the lexicographically smallest path which will take the minimum amount of time (in minutes) required to move from city X to city Y.

It is guaranteed that the given flight route map will be connected. Graph won't contain multiple edges and self loops. A self loop is an edge that connects a vertex to itself.

Input Format:

The first line contains 4 space separated integers, N, M, T and C. Next M lines contains two space separated integers each, U and V denoting that there is a bidirectional road between city U and city V. Next line contains two space separated integers, X and Y.

Output Format:

In the first line print an integer K, denoting the number of city you need to go through to reach city Y from the city X. In next line, print K space separated integers denoting the path which will take the minimum amount of time (in minutes) required by to move from city X to city Y. There can be multiple paths. Print the lexicographically smallest one.

Constraints:

1

≤

N

≤

10

3

N

−

1

≤

M

≤

N

∗

(

N

−

1

)

2

1

≤

C

≤

10

3

1

≤

T

≤

10

3

1

≤

U

,

V

,

X

,

Y

≤

N

SAMPLE INPUT

5 5 3 5

1 2

1 3

2 4

1 4

2 5

1 5

SAMPLE OUTPUT

3

1 2 5

Explanation

Fastest path will be

1

−

>

2

−

>

5

. You can reach city 2 in 5 minutes. After 3 minutes the airport in city 2 will change its state to Loading. So in city 2, you have to wait for 1 minute for the airport to change its state. So total time will be 5 minutes (from city 1 to city 2) + 1 minute (waiting time at city 2) + 5 minutes (from city 2 to city 5) =

11

minutes.

Time Limit: 1.0 sec(s) for each input file.

Memory Limit: 256 MB

Source Limit: 1024 KB

Marking Scheme: Marks are awarded if any testcase passes.

Allowed Languages: Bash, C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Swift, Swift-4.1, TypeScript, Visual Basic

[28/02, 11:54 am] +91 98724 03283: Problem no 2:

In web application development, it is often required to draw a pop-up window on top of a web page when the user clicks on an icon on the page that brings up the pop-up.

You are given a web page with width x and height y, a pop-up window with width l and height m, and an icon at distance a from the left of the page and distance b from the bottom of the page.

Your program should output the orientation in which the pop-up can be rendered fully, relative to the icon, within the page dimensions. The orientation of pop up should be such that it lies completely within the page.

There are 4 possible orientations: bottom-right, bottom-left, top-right and top-left, in the same order of preference. In other words, you should first attempt to render the pop-up bottom-right, and if that is not possible, try bottom-left, and so on.

Note: Icon location and pop-up size are such that the pop-up can always be fully rendered within the page.

Input:

First line contains an integer, T, denoting the number of test cases.

Next T lines contain six space separated integers each, x, y, l, m, a and b.

Output:

For each test case, print the orientation in which the pop-up can be rendered fully, relative to the icon, within the page dimensions.

Constraints:

1

≤

T

≤

10

5

1

≤

x

≤

3000

1

≤

y

≤

3000

1

≤

l

≤

x

1

≤

m

≤

y

0

≤

a

≤

x

0

≤

b

≤

y

There is always an orientation possible such that the pop-up will lie within the page.

SAMPLE INPUT

4

1024 768 200 300 200 600

1024 768 200 300 1000 600

1024 768 200 300 200 200

1024 768 200 300 900 100

SAMPLE OUTPUT

bottom-right

bottom-left

top-right

top-left

[28/02, 11:55 am] +91 98724 03283: These two are specifically for service now .

These are also part of weekly practice

Best wishes

Jagandeep