Spring 2012 Programming Competition

Run by UPE at
Rensselaer Polytechnic Institute

March 31, 2012
12:15 - 2:15

Sponsored By

0. Introduction

Welcome to this semester's programming competition held by UPE. This semester we are lucky enough to be sponsored by Palantir, Microsoft, TripAdvisor, and Vistaprint. To find more details and to submit your solutions please check `progcomp.upe.cs.rpi.edu`.

Please remember our simple rules:

1. All submitted code must be your own.
2. You are not allowed communicate with/ask questions of any other person.
3. You are allowed to use the internet.

Any attempt to tamper with competition server is considered dishonesty with respect to the contest. Please remember you are always subject to RPI's COMEC policy.

0.1 Questions and Submissions

In this document you will find 4 problems in no particular order. The input and output should be handled as you see fit, but we recommend using file input and output as you will be downloading an input file from the server and uploading an output file to the server. You will have two minutes to run your code and submit the output file and program file.

0.2 Input Format

This semester the first line of the input will be a single number `n` which is the number of test cases you will find in that text file. There will then be `n` inputs as defined in the "single input" subsection of each problem.

This means that you should implement the following pseudocode where `RUN SINGLE SOLUTION` reads in the necessary inputs from `F` and then returns the expected output.

```
F = file(input.txt)
Fout = file(output.txt)
n= F.readint()
for i = 1 to n
    print Fout, RUN_SINGLE_SOLUTION(F)
```

Please understand that we will be diffing your output file with one generated by our solution, as such please do not have any extra newlines (you can edit your output by hand if necessary)

1. Intro to Mathematics

Given two integers `n` and `v`, how many additions or subtractions of powers of `v` does it take to equal `n`.

For example for `n`= 10 and `v`= 2 takes one addition.

$$10 = 2^3 + 2^1$$

1.1 Single Input

```
n v
```

1.2 Limits

`number of test cases` $= 500$
$1 \leq$ `n` $\leq 10^9$
$1 \leq$ `v` ≤ 100

1.3 Single Output

```
q
```

`q` is the number of additions and subtractions

1.4 Sample Input

```
4
10 2
8 2
8 3
100 3
```

1.5 Sample Output

```
1
0
1
3
```

2. Maximum strictly monotonically decreasing subsequence

A strictly monotonically decreasing sequence is a sequence in which every term is strictly less than the previous term. $\{16, 12, 8, 2\}$ is such a sequence, but $\{8, 3, 2, 2\}$ and $\{2, 1, 4\}$ are not.

Your challenge is to find the highest sum strictly monotonically decreasing subsequence from a list of numbers. For example, if you were given $\{8, 2, 1, 4, 3\}$ you would chose $\{8, 4, 3\}$ to form a sequence with a sum of 15.

2.1 Single Input

```
n
n space separated integers (v)
```

2.2 Limits

`number of test cases` $= 500$

$1 \leq$`n`≤ 1000

$1 \leq$`v`$\leq 10^6$

2.3 Single Output

```
Sum of best increasing subsequence
```

2.4 Sample Input

```
3
5
14 12 16 8 2
5
8 3 4 1 2
10
5 5 4 4 3 3 2 2 1 1
```

2.5 Sample Output

```
36
14
15
```

3. Experimental Time Travel

Now that Doc has his 1.21 jiggawatt generator back, he has to go back (and forward) in time to fix some mistakes. Of course, he wants to minimize the amount of energy it takes to do this. However, he also can't spend the rest of his life fixing these mistakes.

Given a list of years that he needs to visit, and a maximum time in years he is allowed to spend on the trip. Find the path with the least number of jumps through time such that the time elapsed is less than the maximum time allowed. You should return the number of jumps required.

For this problem, making a jump and fixing mistakes takes no time. The only way time elapses is if you chose to remain (without jumping) at a time until the next year. It takes one year to elapse to the next year. You start outside the timeline, so you must use at least 1 jump.

For example, if the times to visit were 1990, 1991, 2010, and 2013 and we allowed for three years of time. Doc could do this in 3 jumps in two ways.

1. jump to 1990
2. stay until 1991 (1 year elapsed)
3. jump to 2010
4. jump to 2013

He could also do it

1. jump to 1990
2. jump to 1991
3. jump to 2010
4. stay until 2011
5. stay until 2012
6. stay until 2013 (three years elapsed)

Both would be valid since they both return a jump count of 3.

3.1 Single Input

```
k z
k years space separated
```

3.2 Limits

`number of test cases` $= 500$
$1 \leq$ `k` ≤ 1000
$0 \leq$ `z` ≤ 1000
$1000 \leq$ `years` ≤ 9999

You can assume that years are always in increasing order.

3.3 Single Output

```
number of jumps
```

3.4 Sample Input

```
3
4 2
2011 2012 2013 2014
4 3
1990 1991 2010 2013
5 5
1990 1991 1992 2013 2015
```

3.5 Sample Output

```
2
3
2
```

4. Repeating Teleportation

Welcome to 2100 humanoid. Your job is to update the teleportation system put in place 50 years ago. Back then, teleportation only allowed for point to point communication, and it was costly to go over large distances. This led to lots of weird connections which resemble the airports of the early 2000's. Fortunately, a recent invention, the teleportation repeater makes it possible to route traffic so we are no longer limited to point to point communication. Due to the high cost of putting in new teleportation routes however, we would like to utilize the current system but place routers in some cities where necessary.

Given a list of city to city connections and the cost of maintaining them, return the best placement of routers to minimize the maintenance costs on the surviving lines. You must return the expected maintenance cost.

For example:

If we have a three city system

- Chicago to NY 800
- Chicago to LA 1730
- NY to LA 2443

We should place a single router in Chicago and keep the Chicago to NY and Chicago to LA routes. This has a recurring cost of $800 + 1730 = 2530$.

4.1 Single Input

```
n
(next n rows)
Place1 Place2 d
```

4.2 Limits

`number of test cases` $= 25$

$1 \leq$ `n` ≤ 250

$1 \leq$ `d` ≤ 3000

`Places are alpha strings all capitals`

4.3 Single Output

```
c
```

1. `c` is the cost of the system

4.4 Sample Input

```
3
3
```

```
CHICAGO NY 800
CHICAGO LA 1730
NY LA 2443
5
TROY ALBANY 10
ALBANY NY 100
ALBANY BOSTON 300
NY BOSTON 200
ALBANY SARATOGA 20
6
CHI NY 600
CHI BOS 750
CHI LA 1700
CHI SF 1600
LA SF 90
NY BOS 100
```

4.5 Sample Output

```
2530
330
2390
```