Fall 2012 Programming Competition

Run by UPE at Rensselaer Polytechnic Institute

October 28, 2012
12:15 - 2:15

Sponsored By

1 Frequency Detection

One common technique when analyzing encryped data is to look for common sequences of characters. You are given a string of a certain length, and your task is to output a brief summary containing the *three* most frequent 2-character pairs found in the string and how many times it was found occuring. If two or more are tied for the same frequency, sort them alphabetically. Separate the output between each case with three hyphens.

1.1 Single input

```
<string length>
<encrypted string>
```

1.2 Limits

```
<number of test cases> = 200
<string length> ≤ 8000
```

1.3 Single output

```
<first sequence> <number of occurrences>
<second sequence> <number of occurrences>
<third sequence> <number of occurrences>
---
```

1.4 Sample input

```
2
19
ABCDABGHRPIJABDNRPI
22
OPOPOPOPPAGANGNAMSTYLE  (I am so, so sorry)
```

1.5 Expected output

```
AB 3
PI 2
RP 2
---
OP 4
PO 3
AG 1
---
```

2 City Navigation

You live in a city with a street grid, and you need to get across town for class. However, many streets are under construction at this time of the year. Given that it's *only* a humanities class, you decide that you only want to bother leaving if there is a path between you and the classroom with the shortest manhattan distance possible.

Given your starting position, your destination, and some road blocks, find the number of routes you can take that are the smallest manhattan distance to your destination which do not involve taking a street that has construction on it. The location of the starting and ending position is an intersection of two streets as a coordinate pair. A road block is the street between two intersections, which is represented as two coordinate pairs.

2.1 Single input

```
<width> <height>
<start x> <start y> <end x> <end y>
<number of zones>
<zone x1> <zone y1> <zone x2> <zone y2>
<zone x1> <zone y1> <zone x2> <zone y2>
...
```

2.2 Limits

```
<number of test cases> = 200
<grid size> ≤ 100
<number of zones> ≤ 10 000
```

2.3 Single output

```
<number of paths>
```

2.4 Sample input

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

2.5 Expected output

```
20
2
```

3 Password Fragments

An online site for storing bitcoins has been hacked (big surprise), and some interesting details were revealed about their security setup. Firstly, their users were not allowed to use any duplicate characters in their passwords. Not only that, but when logging in, the user only had to input a certain number of characters from their password as long as they kept them in order.

For example, if a user with the password `asdfjkl;` is told to input 3 characters, they may input `asd`, `afl`, `aj;`, etc. As long as the letters are kept in the same order of the password, it will be accepted.

To add to the poor security, all successful logins were recorded to a text file for each user. You happen to stumble upon one such file and wonder if it is possible to reconstruct the actual password with the given fragments. Given a number of login attempts, try to piece them together to restore the full password.

3.1 Single input

```
<number of fragments> <size of each fragment>
<fragment>
<fragment>
...
```

3.2 Limits

```
<number of test cases> = 200
<number of fragments> ≤ 500
<size of each fragment> ≤ 50
```

3.3 Single output

```
<password>
```

3.4 Sample input

```
2
5 3
123
abc
1b3
1c2
1ac
6 4
OGco
ROom
POco
PRop
POGp
Gcmp
```

3.5 Expected output

```
1abc23
PROGcomp
```

4 1D War

We're dealing with the imaginary univerise of dimenionsia, where the entire world is one dimensional. One planet in this universe is in an unstable environment, as all the territories are about to go to war over resources unless you put together a peacekeeping force that can maintain the peace.

There are N territories on the planet, and every territory must have at least 1 peacekeeping force placed on it. Since funding is tight, you'd like to allocate as few troops as possible while still ensuring that no territory will go to war with another.

Each territory has a certain amount of resources, represented as an integer. A territory will only go to war with a neighboring territory, but only if its neighbor has more resources than it has. To prevent this, you simple need to place more peacekeeping forces on the higher-resource territory than you placed on its greedy neighbor.

Hence if two territories are adjacent and have the same amount of resources, then one can have more troops than the other without reason for concern, however if one territory has more resources, it must have more troops.

Calculate the minimum number of troops needed to maintain the peace.

4.1 Single input

```
<number of territories>
<rank of territory>
<rank of territory>
...
```

4.2 Limits

```
<number of test cases> = 200
<number of territories> ≤ 5000
<rank of territory> ≤ 1000
```

4.3 Single output

```
<number of troops deployed>
```

4.4 Sample input

```
2
3
10
100
50
6
13
15
12
8
7
19
```

4.5 Expected output

```
4
13
```