Assignment # 7

April 12, 2009

1 Analyse the functions

Give an example of functions $f(n)$, $g(n)$ such that neither $f(n)$ is $O(g(n))$ nor $g(n)$ is $O(f(n))$. To put it in other words the following propositions hold false for $f(n)$ and $g(n)$.

1. $f(n) = O(g(n))$
2. $g(n) = O(f(n))$

Both of the above propositions are false.

Extra credit:

Characterize the class of functions $f(n)$ and $g(n)$ that satisfy this.

2 Complexity analysis

Compute the time complexity of the code snippet given below in terms of input size n. Note that j is an integer.

```
while(n){
        j=n;
        while(j>1){
                j-=n/(log(j)*log(j));
        }
        n/=2;
}
```

3 Design the Data structure

The operations which the datastructure should support and their complexities are shown in the table below.

insert(x)	$O(log(n))$
extractMedian()	$O(log(n))$
findMedian()	O(1)
buildDataStructure()	O(n)

Please note that

- n is the input size
- extractMedian() removes the median from the set.
- build data structure takes linear time to build the data structure from n elements.

4 Can you see them ?

4.1 Problem Description Goes here

N people are waiting in line to enter a concert. People get bored waiting so they turn and look for someone familiar in the line. Two persons A and B standing in line can see each other if they're standing right next to each other or if no person between them is strictly taller than person A or person B. Write a program that determines the number of pairs of people that can see each other.

4.2 Input Format:

The first line of input contains an integer N ($1 \leq N \leq 500000$), the number of people standing in line. Each of the following N lines contains a single integer, the height of one person in nanometres. Everyone will be shorter than 2^{31} nanometres. The heights are given in the order in which people are standing in line.

4.3 Output Format:

Output the number of pairs of people that can see each other on a single line. Constraints $1 \leq N \leq 500000$, $Height < 2^{31}$

4.4 Sample Input

```
7
2
4
1
2
2
5
1
```

4.5 Sample Output:

```
10
```

5 Prime Path

5.1 Problem Description Goes here

The ministers of the cabinet were quite upset by the message from the Chief of Security stating that they would all have to change the four-digit room numbers on their offices.

- It is a matter of security to change such things every now and then, to keep the enemy in the dark.
- But look, I have chosen my number 1033 for good reasons. I am the Prime minister, you know!
- I know, so therefore your new number 8179 is also a prime. You will just have to paste four new digits over the four old ones on your office door.
- No,it's not that simple. Suppose that I change the first digit to an 8, then the number will read 8033 which is not a prime!
- I see, being the prime minister you cannot stand having a non-prime number on your door even for a few seconds.
- Correct! So I must invent a scheme for going from 1033 to 8179 by a path of prime numbers where only one digit is changed from one prime to the next prime.

Now, the minister of finance, who had been eavesdropping, intervened.

- No unnecessary expenditure, please! I happen to know that the price of a digit is one pound.
- Hmm, in that case I need a computer program to minimize the cost. You don't know some very cheap software gurus, do you?
- In fact, I do. You see, there is this programming contest going on

Help the prime minister to find the cheapest prime path between any two given four-digit primes! The first digit must be nonzero, of course. Here is a solution in the case above.

```
1033
1733
3733
3739
3779
8779
8179
```

The cost of this solution is 6 pounds. Note that the digit 1 which got pasted over in step 2 can not be reused in the last step a new 1 must be purchased.

5.2 Input Format:

One line with a positive number: the number of test cases (at most 100). Then for each test case, one line with two numbers separated by a blank. Both numbers are four-digit primes (without leading zeros).

5.3 Output Format:

One line for each case, either with a number stating the minimal cost or containing the word Impossible.

5.4 Sample Input

```
3
1033 8179
1373 8017
1033 1033
```

5.5 Sample Output:

```
6
7
0
```

source : Taken from http:spoj.pl

6 Help the Soldier

6.1 Problem Description Goes here

Igor, a famous russian soldier, must go to war in Afghanistan (we are in late 80's). His superiors allowed him to buy himself his equipment. So, he must buy 6 items: helmet, bulletproof vest, trousers, boots, tunic and a firearm. This items are represented with numbers from 1 to 6. There are N($6 < N < 101$) items of this 6 types. Each item is characterized by its price p[i] (in rublas) and is quality q[i]. Igor has T ($0 < T < 1001$) rublas and he wants to maximize the total quality of his equipment. The total quality is the quality of the item with the lowest quality. Help him.

6.2 Input Format:

On the first line there are two integers N and T. On the lines 2 ... N+1 there are 3 integers, type[i] (from 1 to 6) p[i] and q[i]. ($0 < p[i], q[i] < T$)

6.3 Output Format:

Output the total quality.

6.4 Sample Input

```
7 53
5 8 2
2 4 8
6 8 13
1 13 12
4 5 1
3 2 7
```

```
3 13 5
```

6.5 Sample Output:

```
1
```

source: From http:spoj.pl

7 Longest Path in the tree

7.1 Problem Description Goes here

You are given an unweighted, undirected tree. Write a program to output the length of the longest path (from one node to another) in that tree. The length of a path in this case is number of edges we traverse from source to destination.

7.2 Input Format:

The first line of the input file contains one integer N — number of nodes in the tree ($0 < N \leq 10000$). Next N-1 lines contain N-1 edges of that tree — Each line contains a pair (u, v) means there is an edge between node u and node v ($1 \leq u, v \leq N$).

7.3 Output Format:

Print the length of the longest path on one line.

7.4 Sample Input

```
3
1 2
2 3
```

7.5 Sample Output:

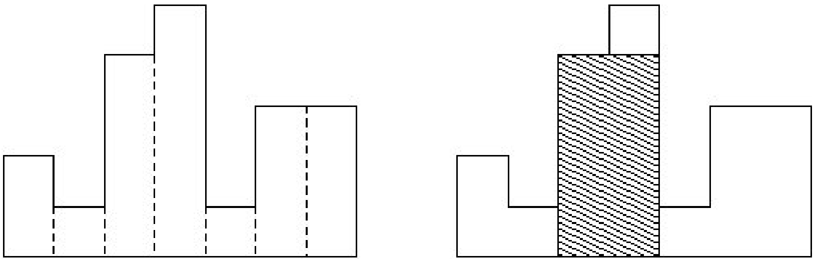
```
2
```

8 Largest Rectangle in a histogram

8.1 Problem Description Goes here

A histogram is a polygon composed of a sequence of rectangles aligned at a common base line. The rectangles have equal widths but may have different heights. For example, the figure on the left shows the histogram that consists of rectangles with the heights 2, 1, 4, 5, 1, 3, 3, measured in units where 1 is the width of the rectangles:

Usually, histograms are used to represent discrete distributions, e.g., the frequencies of characters in texts. Note that the order of the rectangles, i.e., their heights, is important. Calculate the area of the largest rectangle in a histogram that is aligned at the common base line, too. The figure on the right shows the largest aligned rectangle for the depicted histogram.



8.2 Input Format:

The input contains several test cases. Each test case describes a histogram and starts with an integer n, denoting the number of rectangles it is composed of. You may assume that $1 \leq n \leq 100000$. Then follow n integers $h_1, \ldots h_n$, where $0 \leq h_i \leq 1000000000$. These numbers denote the heights of the rectangles of the histogram in left-to-right order. The width of each rectangle is 1. A zero follows the input for the last test case.

8.3 Output Format:

For each test case output on a single line the area of the largest rectangle in the specified histogram. Remember that this rectangle must be aligned at the common base line.

8.4 Sample Input

```
7 2 1 4 5 1 3 3
4 1000 1000 1000 1000
0
```

8.5 Sample Output:

```
8
4000
```

9 Promotion

9.1 Problem Description Goes here

A large Bytelandian supermarket chain has asked you to write a program for the simulating costs of a promotion being prepared.

The promotion has to follow the following rules:

- A customer who wants to participate in the promotion, writes on the receipt, paid by himself, his personal details and throws it into a special ballot box.
- At the end of every day of the promotion, two bills are taken out from the ballot box:
  - first, the receipt amounting to the largest sum is chosen,

  - then the receipt amounting to the smallest sum is chosen;

  The customer who has paid the largest sum gets a money prize equal to the difference between the sum on his bill and the sum on the bill amounting to the smallest sum.

- To avoid multiple prizes for one purchase, both bills selected according to the above rules are not returned to the ballot box, but all remaining bills still participate in the promotion.

The turnover of the supermarket is very big, thus an assumption can be made, that at the end of every day, before taking out receipts amounting to the largest and the smallest sum, there are at least 2 receipts in the ballot box.

Your task is to compute (on the basis of information about prices on receipts thrown into the ballot box on each day of promotion) what the total cost of prizes during the whole promotion will be.

Write a program, which: reads from the standard input a list of prices on receipts thrown into the ballot box on each day of the promotion, computes the total cost of prizes paid in consecutive days of promotion, then writes the result to the standard output.

9.2 Input Format:

The first line of the input contains one positive integer n ($1 \leq n \leq 5000$), which is the duration of promotion in days. Each of the next n lines consists of a sequence of non-negative integers separated by single spaces. Numbers in the (i+1)-th line of the file represent prices on receipts thrown into the ballot box on the i-th day of promotion. The first integer in the line is k, $0 \leq k \leq 10^5$, the number of receipts on the day, and the next k numbers are positive integers standing for the sums on receipts; none of these numbers is larger than 10^6.

9.3 Output Format:

The output should contain exactly one integer, equal to the total cost of prizes paid during the whole promotion.

9.4 Sample Input

```
5
3 1 2 3
2 1 1
4 10 5 5 1
0
1 2
```

9.5 Sample Output:

```
19
```

10 Longest Prefix

10.1 Problem Description Goes here

The structure of some biological objects is represented by the sequence of their constituents denoted by uppercase letters. Biologists are interested in decomposing a long sequence into shorter ones called primitives.

We say that a sequence S can be composed from a given set of primitives P if there is a some sequence of (possibly repeated) primitives from the set whose concatenation equals S. Not necessarily all primitives need be present. For instance the sequence ABABACABAAB can be composed from the set of primitives

$$A, AB, BA, CA, BBC$$

The first K characters of S are the prefix of S with length K. Write a program which accepts as input a set of primitives and a sequence of constituents and then computes the length of the longest prefix that can be composed from primitives.

10.2 Input Format:

First, the input file contains the list (length 1..200) of primitives (length 1..10) expressed as a series of space-separated strings of upper-case characters on one or more lines. The list of primitives is terminated by a line that contains nothing more than a period ('.'). No primitive appears twice in the list. Then, the input file contains a sequence S (length 1..200,000) expressed as one or more lines, none of which exceed 76 letters in length. The "newlines" are not part of the string S.

10.3 Output Format:

A single line containing an integer that is the length of the longest prefix that can be composed from the set P.

10.4 Sample Input

```
A AB BA CA BBC
.
ABABACABAABC
```

10.5 Sample Output:

```
11
```