Data Structures and Algorithms

COSC 336 Assignment 4

Instructions.

1. Due date and time: As announced on Blackboard.

2. This is a team assignment. Work in teams as in the previous assignments. Submit one assignment per team, with the names of all students making the team.

3. Your programs must be written in Java.

4. Write your programs neatly - imagine yourself grading your program and see if it is easy to read and understand. At the very beginning present your algorithm in plain English or in pseudo-code (or both). Comment your programs reasonably: there is no need to comment lines like "i++" but do include brief comments describing the main purpose of a specific block of lines.

5. You will submit on **Blackboard** two files.

   The **first file** is a pdf file (produced ideally with latex and Overleaf) and it will contain the following:

   (a) The solutions to the questions in Exercises 1, 2, 3, and 4.

   (b) A short description of your algorithm for the programming task. Focus on the changes you made compared to the solution given on the indicated web site.

   (c) A table with the results your program gives for the 2 data sets given in input-4.3.txt and input-4.4.txt.

   (d) The java code (so that the grader can make observations).

   The **second file** is the .java file containing the java source code, so that the grader can run your program.

   For editing the pdf file, I recommend that you use Latex, see the template files posted on Blackboard:

   assignment-template.tex and assignment-template.pdf

Exercise 1. For each of the following functions, give a $\Theta(t(n))$ estimation with the simplest possible $t(n)$ (for example $3n^2 + 5n \log n = \Theta(n^2)$).

1. $13n^2 - 2n + 56$
2. $2.5 \log n + 2$
3. $n(12 + \log n)$
4. $1 + 2 + 3 + \ldots + 2n$
5. $1 + 2 + 3 + \ldots + n^2$
6. $\log(n^3) + 10$
7. $\log(n^3) + n \log n$
8. $n \log(n^3) + n \log n$
9. $2^{2 \log n} + 5n + 1$

Exercise 2.

1. Evaluate the following postfix arithmetic expression: $10\ 3\ 4\ -\ 5\ *\ /$
2. Convert the following infix arithmetic expression to postfix notation: $(((2+3)*5)-15)$

Exercise 3.

Consider the following algorithms A and B for the problem of computing $2^n \pmod{317}$ (This is the modular exponentiation problem that we have discussed in class. Algorithm A is the one that we have seen in class, and algorithm B is a variant of it.

```
Algorithm A.

mod_exp_A(n) {
   if (n== 0) return 1;
   else {
          t = mod_exp_A(n/2);
          if (n is even) return t*t (mod 317);
          if (n is odd) return t*t*2 (mod 317);
}}

Algorithm B.

mod_exp_B(n) {
   if (n== 0) return 1;
   else {
          if (n is even) return   mod_exp_B(n/2) * mod_exp_B(n/2) (mod 317);
          if (n is odd) return  mod_exp_B(n/2) * mod_exp_B(n/2) *2 (mod 317);
}}
```

1. Write the recurrence for the runtime $T_A(n)$ of algorithm A, and solve the recurrence to find a $\Theta(\cdot)$ estimation of $T_A(n)$.

2. Write the recurrence for the runtime $T_B(n)$ of algorithm B, and solve the recurrence to find a $\Theta(\cdot)$ estimation of $T_B(n)$.

3. Which algorithm is faster? (Note: There is a huge difference between T_A and T_B.)

Exercise 4. Give a $\Theta(\cdot)$ evaluation for the runtime of the following code:

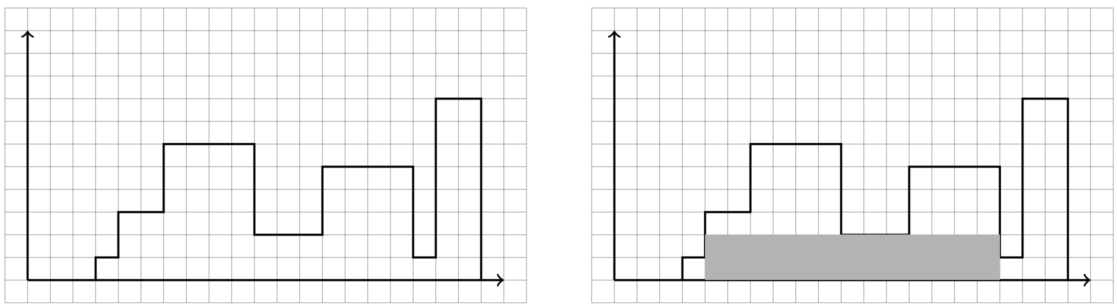
```
 i= 1; x=0;
 while(i <= n) {
        j=1;
        while (j <= i) { x=x+1; j= 2*j; }
        i= 2*i;
}
```

Hint: You can assume that n is a power two. Then i from the outer loop takes successivley the values: $1, 2, 2^2, 2^3, ..., 2^{\log n}$.)

Programming task 1.

Given is a polygon with n vertices $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ consisting of adjacent vertical towers as in the picture below. More precisely, n is even and the line segments alternate between vertical and horizontal (that is, $x_{2i-1} = x_{2i}$ for every $i \in \{1, \ldots, n/2\}$, and $y_{2i} = y_{2i+1}$ for every $i \in \{1, \ldots, n/2 - 1\}$. Additionally, $y_1 = y_n = 0$, every other y-coordinate is positive ($y_i > 0$ for every $y \in \{2, \ldots, n-1\}$), and the x-coordinates form a non-decreasing sequence (that is, $x_1 \le x_2 \le x_3 \le \ldots \le x_n$). Design an $O(n)$ algorithm that finds the largest possible area of an axis parallel rectangle that fits inside this polygon.

For example, the input on the left is given by vertices (3, 0),(3, 1),(4, 1),(4, 3),(6, 3),(6, 6), (10, 6),(10, 2),(13, 2),(13, 5),(17, 5),(17, 1),(18, 1),(18, 8),(20, 8),(20, 0). The rectangle with the largest area that fits inside this polygon is shown on the right, its area is 26.



This task can be done using a stack. See the main idea applied to a simplified version of the problem at

https://www.geeksforgeeks.org/largest-rectangle-under-histogram

You can use the same idea, but you need to make some adjustments because now the towers do not necessarily have width 1. Note that it does not work to break a tower into sub-towers of width 1, because a tower may have a large width, and in this case you obtain a large number of sub-towers which prevents the $O(n)$ solution. In your description (in the pdf file) explain the adjustments you made.

Input specification: the first line contains an even positive integer n, indicating the number of vertices of the polygon. Then n lines follow. The i-th line contains two integers, x_i and y_i, separated by space. You may assume that all numbers fit within int and that n is at most 1,000,000.

Output specification: the output is a single line with the maximum area of an axis-parallel rectangle that fits inside the polygon.

Sample inputs :
input-4.1.txt
input-4.2.txt
Sample outputs :
answer-4.1.txt
answer-4.2.txt

Test your program on the following inputs:
input-4.3.txt
input-4.4.txt
and report the results you have obtained for these two inputs.