CS 341, Spring 2019 A. Jamshidpey, E. Zima

ASSIGNMENT 2

DUE: Friday, June 7, 6 PM. DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read `http://www.student.cs.uwaterloo.ca/~cs341` for general instructions and policies. Also read Assignment section of course outline for clarification of what "justify" means.
Note: All logarithms are base 2 (i.e., $\log x$ is defined as $\log_2 x$).
Note: For all algorithm design questions, you must give the algorithm, argue the correctness, and analyze time complexity.

1 Solving recurrences ...

1.1. Solve the following recurrences by the recursion-tree method (you may assume that n is a power of 3 in part (a) and a power of 5 in part (b)):

(a) [3 marks]

$$T(n) = \begin{cases} 11, & n = 1, \\ 4T(n/3) + n\sqrt{n}, & n > 1. \end{cases}$$

(b) [4 marks]

$$T(n) = \begin{cases} 1, & n = 1, \\ 3T(\frac{3}{5}n) + n^2, & n > 1. \end{cases}$$

1.2.

(a) [1 marks] Solve part (a) of the previous question using Master theorem.

(b) [2 marks] Solve part (b) of the previous question using Master theorem.

1.3. [4 marks] Given recurrence

$$T(n) = \begin{cases} 2, & n < 4, \\ T\left(\lfloor \frac{1}{2}n \rfloor\right) + T\left(\lfloor \frac{1}{4}n \rfloor\right) + n \log n, & n \geq 4, \end{cases}$$

find $f(n)$ such that $T(n) = \Theta(f(n))$.

2 How fast can we find the maximum ...

2.1. [8 marks] Given a sorted in descending order array $X[1..n]$ of distinct integers, the `CLS` operation was applied to it several times. The `CLS` (Circular Left Shift) shifts all entries 1 position to the left with the first element moved to the last entry:

```
temp := X[1]; for i from 1 to n-1 do X[i]:=X[i+1] od; X[n] := temp;
```

Give an algorithm with complexity in $o(n)$ to find the largest value in this array. Justify correctness and analyze running time.

3 This is all about the scheduling ...

3.1. [10 marks] Suppose you have n activities $a_1, a_2, \ldots, a_n$ where each activity a_i is given by a start time, end time pair, $a_i = (s_i, e_i)$ with $e_i > s_i$. You want to schedule these activities into the fewest number of rooms so that the activities scheduled into the same room do not overlap in time. Let the rooms be $R_1, R_2, \ldots$ Design a greedy algorithm to solve this optimization problem. Prove that it always returns an optimal solution. Justify correctness and analyze running time.

4 "Divide and Conquer" with the programming component.

Consider the following problem. There are n towers defined by their integer coordinates x_i and y_i, $i = 1, 2, \ldots, n$. We say that a tower i dominates a tower j if

$$(x_i > x_j) \wedge (y_i > y_j) \vee (x_i < x_j) \wedge (y_i < y_j).$$

Define the *dominance factor* of tower i as the total number of towers dominated by it. We would like to compute the dominance factor of each tower. A straightforward algorithm for this problem is to check the dominance condition for every pair of towers and has complexity $\Theta(n^2)$.
4.1. [8 marks] Design an efficient divide and conquer algorithm for this problem with complexity $\Theta(n \log n)$. Justify the correctness of your algorithm and analyze its running time.
4.2. [20 marks] Implement your algorithm from previous question in `C` or `C++`.

Input and output The input consists of $n + 1$ lines. The first line has one natural number which is n (the number of towers). The following n lines have three numbers each "$ID_i \quad x_i \quad y_i$" which represent the tower ID and its x-y coordinates. The output has to be n lines, with each having two numbers: tower ID and dominance factor of that tower. Furthemore, it should be printed in increasing order of tower IDs. Your program must read from standard input and write to standard output.

Sample input

```
4
12 3 3
10 4 2
13 5 6
7 2 1
```

Sample output

```
7 3
10 2
12 2
13 3
```

Submission guidelines Submit one zip file through Marmoset: `https://marmoset.student.cs.uwaterloo.ca/` which must only contain your code file named towers.c/towers.cpp. You will see two sample public tests, but the main tests will be secret. Your final score will be the score of your best submission. We will compile/run your program in a Linux environment using the command "`gcc -std=c11 towers.c -o`" for `C` or "`g++ -std=c++14 towers.cpp -o`" for `C++`. There is a time limit for each test case, so make sure you have the most efficient implementation.