COMP 3711 – Design and Analysis of Algorithms
2024 Fall Semester – Written Assignment 2
Distributed: 9:00 on September 30, 2024
Due: 23:59 on October 11, 2024

Your solution should contain
(i) your name, (ii) your student ID #, and (iii) your email address
at the top of its first page.

Some Notes:

- Please write clearly and briefly. In particular, your solutions should be written or printed on *clean* white paper with no watermarks, i.e., student society paper is not allowed.
- Please also follow the guidelines on doing your own work and avoiding plagiarism as described on the class home page. ***You must acknowledge individuals who assisted you, or sources where you found solutions.*** Failure to do so will be considered plagiarism.
- The term *Documented Pseudocode* means that your pseudocode must contain documentation, i.e., comments, inside the pseudocode, briefly explaining what each part does.
- Many questions ask you to explain things, e.g., what an algorithm is doing, why it is correct, etc. To receive full points, the explanation must also be *understandable* as well as correct.
- Submit a SOFTCOPY of your assignment to Canvas by the deadline. If your submission is a scan of a handwritten solution, make sure that it is of high enough resolution to be easily read. At least 300dpi and possibly denser.

1. (20 points) You are given n numbers, where n is a positive power of 2. Describe an algorithm that finds the largest and second largest numbers in $n + \log_2 n - 2$ comparisons. Explain the correctness of your algorithm. Show that your number of comparisons is indeed $n + \log_2 n - 2$.

2. (20 points) Let RANDOM$(1, k)$ be a procedure that draws an integer uniformly at random from $[1, k]$ and returns it. We assume that a call of RANDOM takes $O(1)$ worst-case time. The following recursive algorithm RANDOM-SAMPLE generates a random subset of $[1, n]$ with $m \leq n$ distinct elements. Prove that RANDOM-SAMPLE returns a subset of $[1, n]$ of size m drawn uniformly at random.

```
RANDOM-SAMPLE(m, n)
  if m = 0 then
    return ∅
  else
    S ← RANDOM-SAMPLE(m − 1, n − 1)
    i ← RANDOM(1, n)
    if i ∈ S then
      return S = S ∪ {n}
    else
      return S = S ∪ {i}
    end if
    return S
  end if
```

3. (20 points) You are given a set of rectangles $S = \{R_1, R_2, \ldots, R_n\}$ such that all have their bottom sides on the x-axis. This means that each rectangle R_i is specified by a triple (l_i, r_i, h_i), where l_i and r_i are the x coordinates of its left and right sides, and h_i is the height. For simplicity, you may assume that all l_i, r_i, h_i for $i = 1, \ldots, n$ are distinct.

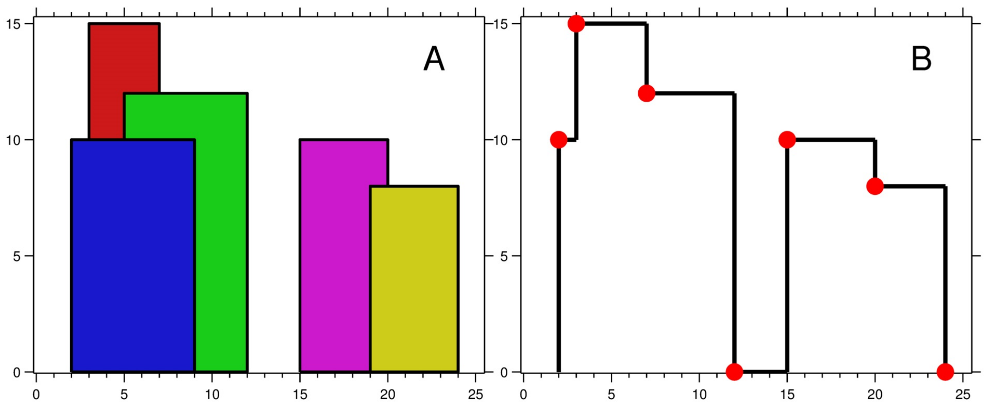
   You will design an algorithm to compute the union $C = R_1 \cup R_2 \cup \cdots \cup R_n$. Clearly, the bottom side of C is a straight line segment from the leftmost lower-left corner u of $R_1, \ldots, R_n$ to the rightmost lower-right corner v of $R_1, \ldots, R_n$. So you only need to find its upper boundary γ. Your algorithm should output γ as a list of "key points" sorted by their x-coordinate in the form $((x_1, y_1), (x_2, y_2), ...)$. Each key point is the left endpoint of some horizontal segment in γ except the last point in the list, which always has a y-coordinate 0 and is used to mark the termination.

   The following figure shows an example, where the input is

   $$((2, 9, 10), (3, 7, 15), (5, 12, 12), (15, 20, 10), (19, 24, 8)),$$

   and the output should be

   $$((2, 10), (3, 15), (7, 12), (12, 0), (15, 10), (20, 8), (24, 0)).$$



Design an algorithm that constructs γ in $O(n \log n)$ time and analyze its running time. Your algorithm MUST use the following divide-and-conquer strategy:

(i) Recursively solve the subproblems for $R_1, \ldots, R_{n/2}$ and $R_{n/2+1}, \ldots, R_n$, respectively. Note that there is no particular ordering among $R_1, R_2, \ldots, R_n$ in the input.

(ii) Let γ_1 and γ_2 be the outputs of the two recursive calls in (i).

(iii) Combine γ_1 and γ_2 to produce γ.

4. (20 points) We explore a different analysis of the application of randomized quicksort to an array of size n.

(a) (2 points) For $i \in [1, n]$, let X_i be the indicator random variable for the event that the ith smallest number in the array is chosen as the pivot. That is, $X_i = 1$ if this event happens, and $X_i = 0$ otherwise. Derive $\mathrm{E}[X_i]$.

(b) (2 points) Let $T(n)$ be a random variable that denotes the running time of randomized quicksort on an array of size n. Prove that

$$\mathrm{E}\big[T(n)\big] = \mathrm{E}\left[\sum_{i=1}^{n} X_i \cdot \big(T(i-1) + T(n-i) + \Theta(n)\big)\right].$$

(c) (2 points) Prove that $E\big[T(n)\big] = \frac{2}{n} \cdot \sum_{i=2}^{n-1} \mathrm{E}\big[T(i)\big] + \Theta(n)$.

(d) (7 points) Prove that $\sum_{k=2}^{n-1} k \log k \le \frac{1}{2} n^2 \log n - \frac{1}{8} n^2$. (Hint: Consider $k = 2, 3, \ldots, (n/2) - 1$ and $k = n/2, \ldots, n-1$ separately.)

(e) (7 points) Use (d) to show that the recurrence in (c) yields $\mathrm{E}\big[T(n)\big] = \Theta(n \log n)$. (Hint: Use substitution to show that $\mathrm{E}\big[T(n)\big] \le cn \log n$ for some positive constant c when n is sufficiently large.)

5. (20 points) Let $A[1..n]$ be an array of n possibly non-distinct integers. The array A may not be sorted. The q-th **quantiles** of $A[1..n]$ are the k-th smallest elements of A for $k = \lfloor n/q \rfloor, \lfloor 2n/q \rfloor, \ldots, \lfloor (q-1)n/q \rfloor$. Note that the q-th quantiles consist of $q-1$ elements of A.

For example, if $A = [5, 8, 16, 2, 7, 11, 0, 9, 3, 4, 6, 7, 3, 15, 5, 12, 4, 7]$, the 3rd quantiles of A are $\{4, 7\}$, because the 3rd quantiles consist of the 6-th and 12-th smallest elements of A, which are 4 and 7, respectively.

Suppose you have a black box worst-case linear-time algorithm that can find the median of an array of integers. That is, this algorithm runs in $O(s)$ time on an array of size s. Describe an algorithm that determines the q-th quantiles of $A[1..n]$ in $O(n \log q)$ time. Argure that your algorithm is correct. Derive the running time of your algorithm.