CSc 220: Algorithms
Homework 2
Due in Class on Thursday September 15

Return the homework written on sheet(s) of paper with your name and CSc220 written at the top of each sheet. Please staple multiple sheets together. Remember that collaboration is allowed, but that you must write the solution on your own. Also you must acknowledge all collaborators and all sources (other than the textbook) in your solutions. Each problem is worth 10 points.

Problem 1: Given a set \( A \) of \( n \) distinct integers we want to find the \( m \)th smallest element of \( A \).

- Give a deterministic \( \Theta(n \log n) \) algorithm to find such element \([2pts]\);
- Give a randomized algorithm that finds it in expected \( O(n) \) time. \([8pts]\)

Later in the class we will show a deterministic \( O(n) \) algorithm to find the \( m \)th element, which is described in the textbook. You will not get credit by presenting that algorithm as a solution for this homework.

Problem 2: A different way to randomize QUICK-SORT is to use the deterministic version of QUICK-SORT over a 'randomized' array, according to the following pseudo-code

\[
\text{PERMUTE-QUICK-SORT}(A) \\
B \leftarrow \text{RANDOM-PERMUTE}(A); \\
\text{RETURN QUICK-SORT}(B)
\]

- Under what conditions on the procedure RANDOM-PERMUTE will PERMUTE-QUICK-SORT run in \( O(n \log n) \) steps? \([2pts]\)
- Consider the following procedure

\[
\text{BLOCK-PERMUTE}(A) \\
n \leftarrow |A|; \\
s \leftarrow \text{RANDOM}(1, n); \\
\text{FOR } i = 1 \text{ TO } n - s \\
\quad B[i] \leftarrow A[s + i]; \\
\text{FOR } i = 1 \text{ TO } s \\
\quad B[n - s + i] \leftarrow A[i]; \\
\text{RETURN } B
\]

What is the expected running time of PERMUTE-QUICK-SORT if you use BLOCK-PERMUTE in place of RANDOM-PERMUTE? \([4pts]\)

- Give your own implementation of RANDOM-PERMUTE that will make PERMUTE-QUICK-SORT run in \( O(n \log n) \) steps. \([4pts]\)

Problem 3: At the end of the academic year CUNY will issue a sorted list of all its Computer Science students, ranked by their score on the Algorithms course. For each section of the course, instructors have been asked to return a sorted list of the students in that section according to their score. At the end of the year CUNY will have \( k \) sorted lists (one for each section), with \( n \) students in total. Show how to produce the total sorted list in \( O(n \log k) \) steps.