CSCI3160: Quiz 1

Name: Student ID

Problem 1 (20%) Prove or disprove: if function $f(n) = O(n \log n)$ and function $g(n) = O(\sqrt{n})$, then $f(n) + g(n) = O(n \log n)$.

If you think the above statement is correct, provide a proof. Otherwise, provide a counterexample.

Solution. The statement is correct. Since $f(n) = O(n \log n)$, there exist constants c_1, c'_1 such that $f(n) \le c_1 \cdot n \log_2 n$ for all $n \ge c'_1$. Since $g(n) = O(\sqrt{n})$, there exist constants c_2, c'_2 such that $g(n) \le c_2 \cdot \sqrt{n}$ for all $n \ge c'_2$. Therefore, $f(n) + g(n) \le (c_1 + c_2) n \log_2 n$ for all $n \ge \max\{c'_1, c'_2\}$.

Problem 2 (40%). Consider an array storing $n = 9$ integers: $A = (50, 20, 40, 60, 80, 90, 10, 30, 70)$. Recall that, in the k-selection algorithm, we randomly select a pivot p from A and then divide A into two arrays:

- A_1, which includes all the elements of A less than or equal to p;
- A_2, which includes all the elements of A greater than p;

After that, we recurse into a subproblem if the subproblem has size at most $2n/3$, or declare "failure" otherwise. Let us set $k = 5$ (i.e., the goal of k-selection is to find the 5-th smallest element in A).

Answer the following questions:

1. If the pivot p equals 40, what is the input to the subproblem?
2. Which of the elements in A will induce failure, if they are selected as p?
3. If p is selected from A uniformly at random, what is the probability we declare failure?

Solution.

1. $(50, 60, 80, 90, 70)$ (ordering does not matter).
2. $10, 20, 70, 80, 90$
   (half marks given if 70 is missing)
3. 5/9 (full marks given as long as the answer is consistent with the answer for question 2)

Problem 3 (40%). Consider running the "counting inversion" algorithm on the array $A = (50, 20, 40, 60, 80, 10, 30, 70)$. Recall that the algorithm divides A into two equal halves at the middle, and recursively solves the subproblems corresponding to the two halves, respectively. Answer the following questions:

1. What are the outputs of the two subproblems, respectively?
2. After recursion, the algorithm will count the number of "crossing inversions". How many crossing inversions are there in A?
3. In the class, we used an $O(n \log n)$-time method to count the number of crossing inversions and proved that the whole algorithm ran in $O(n \log^2 n)$ time. Assume that Mr. Goofy decides to replace our $O(n \log n)$-time method with his own method that runs in $O(n^2)$ time. What is the worst-case time of the whole algorithm now? You need to explain the derivation of your answer.

Solution.

1. 2 and 3
2. 7
3. $f(n) = 2 \cdot f(n/2) + O(n^2)$, which solves to $f(n) = O(n^2)$ (Master Theorem).