CSCI 6470 Quiz #3 Questions (Answers)

September 25, 2023 (11:40am-12:10pm EST)

Student Name _____ **Student ID**_____

Rules. Violation will result in **zero credit** for the exam and possibly the final grade.
1. Closed book/note/electronics/neighborhood.
2. Surrender your cell phone to the podium before using the restroom.

There are 4 questions and 60 points in total. Good luck!

1. (**15 points**) Consider the following recursive algorithm `Work`.

```
function Work(L, i, j); // L is a list, i and j are two indexes;
  if (i <j )
     swap(L[i], L[j]);
     Work(L, i+1, j-1);
  return;               // if such a statement is preferred.
```

   (1) What does algorithm `Work` do? reverse a list [5 *points*]

   (2) If $T(n)$ is the worse case time complexity for `Work` on input of size n, express n in terms quantities i and j only: $n = j - i + 1$ [3 *points*]

   (3) Formulate $T(n)$ in a recursive form, including the base case(s):

$$T(n) = \begin{cases} a \text{ [1 points]} & \text{base case(s) when n =0, 1 [2 points]} \\ T(n-2) + b \text{ [4 points]} & \text{recursive cases when } n \geq 1 \text{ [1 points]} \end{cases}$$

2. (**15 points**) This question concerns the time complexity $T(n)$ of `Quick Sort` based on various scenarios of the selected pivot. Give a recursive formulation for $T(n)$, for $n > 1$,

   (1) On worst case pivot $T(n) = T(n-1) + bn$ [5 *points*]

   (2) On ideal case pivot $T(n) = 2T(n/2) + bn$ [5 *points*]

   (3) On averaged case pivot $T(n) = (n-1) + \frac{2}{n}\sum_{i=0}^{n-1} T(i)$ [5 *points*]

3. (**15 points**) Let list [30, 70, 50, 20, 80, 40, 60] be an input to the `randomized Quick Sort`.

   (1) If the pivot selected by the randomized algorithm is [60], then `Partition` function on this list should return [30, 50, 20, 40, 60, 70, 80] as result; 5 *points*

   (2) If the algorithm is re-run on the same list [30, 70, 50, 20, 80, 40, 60] with pivot [20], will the number of comparisons "used by `Partition` only" be the same as in (1)? Y 3 *points*

   (3) Element 50, 2 *points* if selected as the pivot, yields the ideal situation. Then `Partition` on the same list with **this** pivot should return [30, 20, 40, 50, 80, 60, 70] as result 5 *points*

4. (**15 points**) This question concerns `Selection` algorithm. On the input (A, n, k), the algorithm finds the k^{th} smallest element in set A containing n elements. Specifically, it first finds a pivot x and then uses x to partition set A into two subsets A_1 and A_2 so that the k^{th} smallest element is recursively found from either of the subsets.

   (1) To find pivot x, the algorithm recursively calls `Selection`$(M, \frac{n}{5}, \frac{n}{10})$, where M 2 *points*

   is the collection of medians in groups 1 *points* and x is the median of M 1 *points*

   .

   (2) If rank$(x) = r > k$, recursive call `Selection`$(A_1, r-1, k)$ is executed; 2 *points*

   i f $r < k$, call `Selection`$(A_2, n-r, k-r)$ is executed instead; 3 *points* (2 for $k-r$)

   (3) `Selection` algorithm has its time complexity $T(n)$ formulated in recursive form as

$$T(n) = T(\frac{n}{5}) + T(\frac{7n}{10}) + bn$$

   where $\frac{n}{5}$ is size of M 1 *points* and $\frac{7n}{10}$ is size upper bound for A_1 and A_2 3 *points*

   .

   If the algorithm uses groups of 3 elements instead of 5, the formulation for $T(n)$ would be

$$T(n) = T(\frac{n}{3}) + T(\frac{2n}{3}) + dn, \text{ for some } d > 0$$ 2 *points*

[The following space will not be graded.]