Homework Assignment No. 2

CSCI 4470/6470 Algorithms, CS@UGA, Spring 2019

Due Monday February 11, 2018

Five questions with total 140 points.

Homework submission should be in hardcopy. If for a reason an email submission is necessary, it needs an approval from this instructor. Submission needs to be received by 5:00pm on the due day.

1. (40 points) Consider the data structure heap used in the HeapSort algorithm. For this question, we define the height of a heap to be the number of nodes on the longest path connecting the root and a leaf. Also we define $l(x)$, layer of node $x$ in the heap, recursively as follows.

   - For root $r$, $l(r) = 1$;
   - For any node $x$ with non-empty parent $x.p$, $l(x) = l(x.p) + 1$;

(a) If the heap contains $n$ elements, what is the height of the heap?

(b) How many nodes are there at layer $k$ in the heap, for any given $k \geq 1$?

(c) What is the worst-case number of comparisons used by MAX-HEAPIFY($A, i$) when node $i$ is at the $k^{th}$ layer?

(d) The subroutine BUILD-MAX-HEAP calls subroutine MAX-HEAPIFY($A, i$) $\frac{n}{2}$ times. Use your answers to (1), (2) and (3) to prove that the worst-case time complexity for subroutine BUILD-MAX-HEAP is $O(n)$, which actually is better than the upper bound $O(n \log_2 n)$ shown in the lecture note.

2. (40 points) Consider that duplicate elements may occur in a list to be sorted by Quick Sort.
(a) Modify the pseudo code of function PARTITION so that all elements the same as the pivot will not be considered in subsequent rounds of recursive calls.

(b) Argue that the worst case running time of the modified QUICK SORT is $O(kn)$ if there are only $k$ distinct elements in the input list.

3. (20 points) Given the same scenario described in Q2, i.e., the list to be sorted by the modified QUICK SORT has only $k$ distinct elements out of total $n$ elements. Which of the following upper bounds is best for the average case time complexity of the modified QUICK SORT algorithm? Explain logically.

   (a) $O(n \log_2 n)$   (b) $O(k \log_2 n)$   (c) $O(n \log_2 k)$   (d) $O(k \log_2 k)$

4. (20 points) Use decision tree to prove that searching a sorted list of $n$ elements has the lower bound $\Omega(\log_2 n)$ with the comparison based model. Give a formal proof.

5. (20 points) Consider the SELECT that finds the $k^{th}$ smallest element in a given set. Argue that if, instead of making groups of 5 elements, we make groups of 3 elements, the analysis of the algorithm does not lead to the conclusion of the linear time complexity.

The answers must be word-processed or typed. You may substitute formulae and figures with hand-writings. Your submitted algorithms should be in the pseudo-code, not in any specific programming language. Answers deviating from these requirements will be returned without grading.

The answers must be the student’s own work. Idea sharing and referencing to others’ work (including those online) are not allowed. Plagiarism and other forms of academic dishonesty will be handled within the guidelines of the Student Handbook and reported to the University.