CS575 Homework 2

Instructor: KD Kang

Upload a scanned soft copy in pdf format through Blackboard by 9am on Feb. 27. Submit a hard copy at the beginning of the class on Feb. 27. Please be timely. No late submission will be accepted for this homework.

Please remember to include the following paragraph at the beginning of your submitted assignment, write down Your Name, Section Number, and SIGN it.

"I have done this assignment completely on my own. I have not copied it, nor have I given my solution to anyone else. I understand that if I am involved in plagiarism or cheating I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of 0 for the involved assignment for my first offense and that I will receive a grade of "F" for the course for any additional offense."

1. You are given an input array of integers: [8, 3, 7, 1, 2, 10, 5].

   (a) [4 points] Build a **max**-heap for the input data. (No credit will be given for a min-heap.) Draw figures to show the process.

   (b) [6 points] Using the max-heap built above, sort the input data in non-ascending order using the heap sort algorithm. Draw figures to show every step.

2. Assume that you are given an array of n elements sorted in non-descending order where $n \geq 1$. Given the assumption, solve the following problem via divide and conquer.

   (a) [10 points] Design a ternary search algorithm that searches the array by dividing it into three subarrarys where each sublist has approximately n/3 data elements. Write a pseudo code to do ternary search with the worst case time complexity of $\Theta(\log_3 n)$. (Hint: Extend the binary search algorithm.)

   (b) [10 points] Prove that the worst case time complexity of your algorithm is $\Theta(\log_3 n)$. Write recursive equations and solve them iteratively. (Don't use the master theorem).

3. Use the radix sort algorithm to sort the following list of numbers in non-descending order: 321, 38, 15, 3, 9, 82, 10, 11.

   (a) Draw three figures (one for each digit) to show the process step by step [5 points].

   (b) Repeat part (a) by sorting the most significant digit first. Does it sort the numbers properly? Show the process. [5 points].

4. Prove that the lower bound of sorting based on comparions is $\Omega(nlgn)$ [10 points].

5. The worst case time complexity of quick sort is $O(n^2)$. Regarding this, answer the following questions.

   (a) Briefly describe when the time complexity of quick sort becomes $O(n^2)$ [2 points].

(b) Write recursive equations for the worst case and solve them to prove that the time complexity is $O(n^2)$ [6 points].

(c) Briefly describe how to avoid the worst case in quick sort [2 points].

6. Is it a good idea to apply the divide and conquer method to compute a Fobonacci number?

   (a) Simply say yes or no [1 point].

   (b) Justify your yes/no answer [9 points].

7. Prove the following.

   (a) [10 points] $n^k = o(2^k)$.

   (b) [10 points] $lgn = o(n)$.

   (c) [10 points] $n = o(n^3)$.