CSE 321 Homework 2

Due date: 09 / 11 / 2018

1-) Suppose you are choosing between the following three algorithms:

a) Algorithm A solves problems by dividing them into five subproblems of half the size, recursively solving each subproblem, and then combining the solutions in linear time.
b) Algorithm B solves problems of size n by recursively solving two subproblems of size n - 1 and then combining the solutions in constant time.
c) Algorithm C solves problems of size n by dividing them into nine subproblems of size n/3, recursively solving each subproblem, and then combining the solutions in $O(n^2)$ time.

What are the running times of each of these algorithms (in big-O notation), and which would you choose?

2-) Solve the following recurrence relations and give a Θ bound for each of them.

a) $T(n) = 2T(n/3) + 1$
b) $T(n) = 5T(n/4) + n$
c) $T(n) = 7T(n/7) + n$
d) $T(n) = 9T(n/3) + n^2$
e) $T(n) = 8T(n/2) + n^3$
f) $T(n) = 49T(n/25) + n^{3/2} \log n$
g) $T(n) = T(n - 1) + 2$
h) $T(n) = T(\sqrt{n}) + 1$

3-) Given an array of n elements, and you notice that some of the elements are duplicates; that is, they appear more than once in the array. Show how to remove all duplicates from the array in time O(n log n).

4-) Consider the task of searching a sorted array A[1 ,…, n] for a given element x: a task we usually perform by binary search in time O(log n). Show that any algorithm that accesses the array only via comparisons (that is, by asking questions of the form “is $A[i] \leq z$?”), must take Ω(log n) steps.

5-) How many lines, as a function of n (in Θ (.) form), does the following program print? Write a recurrence and solve it. You may assume n is a power of 2.

```
function f(n)
  if n > 1:
    print_line(‘‘still going’’)
    f(n/2)
    f(n/2)
```

Note:

* Your submissions will be handwritten

* You can deliver your homework to TA Burak Koca until 17:00 on due date (room 119).

* Do your homework personally, group studies will be considered as cheating.