CSE 321 Homework 2

Due date: 11 / 11 / 2019

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

a) $T(n) = 27T(n/3) + n^2$
b) $T(n) = 9T(n/4) + n$
c) $T(n) = 2T(n/4) + \sqrt{n}$
d) $T(n) = 2T(n/2) + 17$
e) $T(n) = 2T(\sqrt{n}) + 1$
f) $T(n) = 4T(n/2) + n$
g) $T(n) = T(n/3) + T(2n/3) + O(n)$
h) $T(n) = T(n - 1) + n^c$, where $c>0$ and c is a constant
i) $T(n) = T(n - 1) + c^n$, where $c>0$ and c is a constant

2-) A binary tree is considered as full when all of its vertices have either zero or two children. Let B_n denote the number of full binary trees that have n vertices.

a) By drawing out all full binary trees with 3, 5, or 7 vertices, determine the exact values of B_3, B_5, and B_7. Why have we left out even numbers of vertices, like B_4?

b) For general n, derive a recurrence relation for B_n.

c) Calculate average-case Θ() complexity of the recurrence relation that is derived on the option b.

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

a) Algorithm A solves problems by dividing them into seven subproblems that have one-third of the size, recursively solving each subproblem, and then combining the solutions in quadratic time.
b) Algorithm B solves problems of size n by recursively solving two subproblems of size n - 1 and then combining the solutions in linear time.
c) Algorithm C solves problems of size n by dividing them into four subproblems of half of the size, 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?

4-) The MINIMUM CUT (Min-Cut) problem is the following: given in input an undirected graph $G = (V, E)$, we want to find the subset $A \subseteq V$ such that $A \neq \emptyset$, $A \neq V$, and the number of edges with one endpoint in A and one endpoint in $V - A$ is minimized. Give a polynomial-time algorithm to find a global min-cut in an undirected graph G. Explain your algorithm and analyze the worst-case, best-case and average-case time complexity of the algorithm.

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

```
function f(n):
    res=0
    if $n \leq 1$:
        res 1
    else:
        for i in range (n):
            res += f(i) * f(n – i – 1)
    print (res)
    return res
```

Note:

* Your submissions must be handwritten or Latex paper. In both way, you must deliver your homework as hardcopy.

* 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.