CS 3510: Design & Analysis of Algorithms (Sections A & B)

Practice Quiz 1

Professors Dana Randall and Gerandy Brito

1.) (20 points total) For each of the following, **select only one option** that most accurately describes the relationship between $f(n)$ and $g(n)$.

(a.) (5 points) $f(n) = n^3 + 2n, \;\; g(n) = 12n^2 + 24\sqrt{n}$

- ○ **A.** $f(n) = \mathcal{O}(g(n))$
- ○ **B.** $g(n) = \mathcal{O}(f(n))$
- ○ **C.** $f(n) = \mathcal{O}(g(n))$ **and** $g(n) = \mathcal{O}(f(n))$

(b.) (5 points) $f(n) = (\log n)^2, \;\; g(n) = (\log n) + \sqrt{n}$

- ○ **A.** $f(n) = \mathcal{O}(g(n))$
- ○ **B.** $g(n) = \mathcal{O}(f(n))$
- ○ **C.** $f(n) = \mathcal{O}(g(n))$ **and** $g(n) = \mathcal{O}(f(n))$

(c.) (5 points) $f(n) = 2\log_5(3^n), \;\; g(n) = n + 4n^{0.2}$

- ○ **A.** $f(n) = \mathcal{O}(g(n))$
- ○ **B.** $g(n) = \mathcal{O}(f(n))$
- ○ **C.** $f(n) = \mathcal{O}(g(n))$ **and** $g(n) = \mathcal{O}(f(n))$

(d.) (5 points) $f(n) = 8^{\log_7 n}, \;\; g(n) = n \log n$

- ○ **A.** $f(n) = \mathcal{O}(g(n))$
- ○ **B.** $g(n) = \mathcal{O}(f(n))$
- ○ **C.** $f(n) = \mathcal{O}(g(n))$ **and** $g(n) = \mathcal{O}(f(n))$

2.) (20 points) Suppose you've discovered two schemes you could use to design a divide and conquer algorithm for a problem.

(A.) Divide a problem of size n into 4 subproblems of size $n/8$, where the cost of combining the subproblem solutions is $\mathcal{O}(\sqrt{n})$.

(B.) Divide a problem of size n into 3 subproblems of size $n/3$, where the cost of combining the subproblem solutions is $\mathcal{O}(n)$.

Which scheme do you prefer and why? Justify your answer.

Algorithm 1 Neil's "Fun" algorithm

```
procedure FUN(n)
    if n = 1 then
        PRINT("yay!")
        return
    for i ← 1 to n do
        for j ← 1 to n do
            PRINT("nay!")
    FUN(n/2)
    FUN(n/2)
    return
```

3.) (25 points total) Suppose we have the following algorithm.

(a.) (10 points) Give a recurrence relation for $T(n)$, the number of lines printed by FUN as a function of its input n.

(b.) (10 points) Solve the recurrence relation. Give your answer in Big-O notation. Show all work, including values of a, b, and d if you use the master theorem.

(c.) (5 points) Of these printed lines, how many are "yay!"s and how many are "nay!"s?

4.) (35 points total) You are given a **sorted** array of **distinct** integers $A = [a_1, a_2, \ldots, a_n]$, which may contain negative values, and you want to find out whether there is an index i such that $A[i] = i$. In this problem, the array A is 1-indexed.

For example, in the array $A = [-3, -2, 0, 3, 4, 6, 9, 11]$, you should return 6, since $A[6] = 6$.

(a.) (20 points) Give a divide-and-conquer algorithm for this problem with running time strictly faster than linear (e.g., $\mathcal{O}(\sqrt{n})$ or $\mathcal{O}(\log n)$). Assume that you can look up the value of $A[i]$ in $\mathcal{O}(1)$ time. Explain your algorithm **and** give pseudocode.

(b.) (10 points) What is the running time of your algorithm? Show all your work.

(c.) (5 points) Briefly argue why your algorithm is correct.