CMPSC 465 — Data Structures & Algorithms
Fall 2024 — Mingfu Shao and Debarati Das

HW 1

Due September 6th (Friday), 11:59 pm

Formatting: Each problem should begin on a new page. When submitting in Gradescope, try to assign pages to problems from the rubric as much as you can. Make sure you write all your group members' names. For the full policy on assignments, consult the syllabus.

0. (0 pts.) Acknowledgements. List any resources besides the course material that you consulted in order to solve the assignment problems. If you did not consult anything, write "I did not consult any non-class materials." The assignment will receive a 0 if this question is not answered.

1. (16 pts.) Asymptotic Growth

In each of the following situations, indicate whether $f(n) = O(g)$, $f(n) = \Omega(g)$, or $f(n) = \Theta(g)$.

- $f(n) = n^3 + n^2, \quad g(n) = n^3$
- $f(n) = n \log n, \quad g(n) = (\log n)^3$
- $f(n) = 2^n, \quad g(n) = 2^{n-1}$
- $f(n) = n^3 2^n, \quad g(n) = 3^n$
- $f(n) = \sqrt{n}, \quad g(n) = \log n$
- $f(n) = \log(n!), \quad g(n) = n \log n$
- $f(n) = n!, \quad g(n) = 2^n \cdot n^2$
- $f(n) = n^{\log \log n}, \quad g(n) = \log(n^{\log n})$

2. (17 pts.) Let f, g be any functions from $\mathbb{N}$ to $(0, \infty)$. Then prove or disprove the following:

- (3 pts) $f = o(g) \Rightarrow f = O(g)$
- (3 pts) $f = O(g) \lor f = \Omega(g)$
- (3 pts) $f = O(g) \land f \neq \Theta(g) \Rightarrow f = o(g)$
- (4 pts) Let $\mathcal{F}$ denote the set of all functions from $\mathbb{N}$ to $(0, \infty)$. Then Θ defines an equivalence relation on $\mathcal{F}$. *(A relation $\mathcal{R}$ on a set S is an equivalence relation if it is reflexive, symmetric and transitive.)*
- (4 pts) For any increasing sublinear function $t : (0, \infty) \to (0, \infty)$, $f = O(g) \Rightarrow t \circ f = O(t \circ g)$. We define t is *sublinear* if for all $\alpha > 0$, $x > 0$, we have $t(\alpha x) \leq \alpha t(x)$. Also, $t \circ f$ is a function defined as $(t \circ f)(n) = t(f(n))$.

3. (15 + 5 (bonus) pts.) For each pseudo-code below, give the asymptotic running time in Θ notation. You may assume that standard arithmetic operations take $\Theta(1)$ time. The 4th part is a bonus part.

1.

```
k := 0;
for i := 1 to n do
    for j := i to n do
        k := k + 1;
    end
end
```

2.

```
for i := 1 to n do
    for j := 1 to n do
        k := j;
        while k ≥ 1 do
            k := k − 1;
        end
    end
end
```

3.

```
for i := 1 to n do
    j := 1;
    while j ≤ n do
        j := j × 2;
    end
end
```

4.

```
for i := 1 to n do
    j := i;
    while j ≤ n do
        j := j + i;
    end
end
```

4. (15 pts.) Master's Theorem

Solve each of the following recurrences. Give the closed form of $T(n)$ in Θ-notation. You may assume that n is of a special form (e.g., a power of two or another number) and that the recurrence has a convenient base case that is $T(1) = \Theta(1)$.

- (3 pts) $T(n) = 4 \cdot T(n/4) + n$
- (3 pts) $T(n) = 6 \cdot T(n/3) + n^2$
- (3 pts) $T(n) = 4 \cdot T(n/8) + \log n$
- (3 pts) $T(n) = 2 \cdot T(n/2) + n \log n$
- (3 pts) $T(n) = 8 \cdot T(n/3) + 2^n$