Advanced Algorithms — Exercise Set 0

Name: ____________________________

- Submit in class on **January 27, 2026**.
- Feel free to discuss with others, but write up your own solutions.
- This will be graded for completion. Use it to learn!

Asymptotics

Big-O notation is used to capture asymptotic upper bounds: if $f(n) = O(g(n))$, we say that f grows no faster than g asymptotically. In other words, it's the asymptotic version of the $\leq$ sign. Similarly, if $f(n) = O(g(n))$, then $g(n) = \Omega(f(n))$. This is pronounced "Big-Omega", and is the asymptotic $\geq$ sign. When both $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$ hold, we write $f(n) = \Theta(g(n))$, indicating that f and g grow at the same asymptotic rate. We use lowercase versions (o and ω) to express strict inequalities.

You may want to use the above notation for the following problem.

Problem 1 (Asymptotics)**.** *Sort the following functions in increasing order of asymptotic growth rate (from slowest to fastest). Be sure to note any asymptotic equalities.*

$$n^2, \quad \log(n^2), \quad 2^n, \quad n\log n, \quad n^{\log n}, \quad \sqrt{n}, \quad \log n, \quad 3^n$$

Which of these terms is at most a polynomial in n?

Do you have any questions asymptotic notation that I can help to clarify?

Graph Theory

We will be working with graphs a lot. I personally love graphs because they crop up everywhere, in theory and in real life. In fact, before I was an algorithms researcher, I just studied graphs.

Problem 2 (Graph Basics)**.** *Let $G = (V, E)$ be an undirected graph.*

(a) Prove that the sum of the degrees of all vertices in a graph is equal to $2|E|$. This is sometimes called the Handshaking Lemma.

(b) Use part (a) to prove that the average degree of vertices in a tree is at most 2.

(c) What is the maximum number of edges in a simple undirected graph with n vertices?

(d) Define what it means for a graph to be connected.

Problem 3 (DFS and BFS)**.** *Briefly describe the difference between depth-first search and breadth-first search. What can each algorithm be used for? If I want to find the distance between u and v in an unweighted graph, what algorithm should I use?*

Problem 4 (Matchings)**.** *Let $G = (V, E)$ be an undirected graph. A set of edges $M \subseteq E$ is called a* ***matching*** *if no two edges in M share an endpoint. A* ***maximum matching*** *in G is a matching of largest size.*

(a) Draw a picture of a connected graph with a matching M of size $n/2$. This is called a ***perfect matching****.*

(b) Now, draw a graph with the most number of vertices you can while still having maximum matching size 1.

What questions do you have about the topics above?

Complexity Classes

In this class, we will learn about and design algorithms for problems which we strongly believe no efficient algorithm exists. First, let's recall some fundamentals.

Problem 5 (P vs NP)**.** *Define the complexity classes* **P** *and* **NP**. *Give an example of a problem in each. What does it mean for a problem to be* **NP***-hard? What about* **NP***-complete? Can you give an example of an* **NP***-complete problem which is not in* **P***? (be careful, that last one might be a trick question).*

Problem 6 (Reduction Intuition)**.** *Suppose that $A \leq_p B$, meaning that there is a polynomial-time reduction from decision problem A to decision problem B. Which of the following implications are always true? Include a brief justification.*

(a) If $B \in \mathbf{P}$, then $A \in \mathbf{P}$.

(b) If $A \in \mathbf{P}$, then $B \in \mathbf{P}$.

(c) If $A \in \mathbf{NP}$, then $B \in \mathbf{NP}$.

Problem 7 (Decision vs Optimization)**.** *Give an example of a decision problem and the corresponding optimization version. Explain the relationship between the two.*

What questions do you have about complexity classes?