

# Discrete Structures (CSCI 246)

## Homework 5.1

---

### Purpose & Goals

The following problems provide practice relating to:

- asymptotic analysis (Big O),
- properties of Big O,
- algorithm analysis,
- the problem solving process.

### Submission Requirements

- **Type or clearly hand-write your solutions** into a pdf format so that they are legible and professional. Submit your pdf to Gradescope. **Illegible, non-pdf, or emailed solutions will not be graded.**
- Each problem should start on a new page of the document. When you submit to Gradescope, associate each page of your submission with the correct problem number. Please post in Discord if you are having any trouble using Gradescope.
- Try to model your formatting off of the proofs from lecture and/or the textbook.
- Submit to Gradescope early and often so that last-minute technical problems don't cause you any issues. Only the latest submission is kept. Per the syllabus, assignments submitted within 24 hours of the due date will receive a 25% penalty and assignments submitted within 48 hours will receive a 50% penalty. After that, no points are possible.

### Academic Integrity

- You may work with your peers, but **you must construct your solutions in your own words on your own.**
- Do not search the web for solutions or hints, post the problem set, or otherwise violate the course collaboration policy or the MSU student code of conduct.
- Violations (regardless of intent) will be reported to the Dean of Students, per the MSU student code of conduct, and you will receive a 0 on the assignment.

### Tips

- Answer each problem to the best of your ability. Partial credit is your friend!
- Read the hints for where to find relevant examples for each problem.
- Refer to the [problem solving and homework tips guide](#).
- It is not a badge of honor to say that you spent 5 hours on a single problem or 15 hours on a single assignment. Use your time wisely and get help (see "How to Get Help" below).

## How to Get Help

When you are stuck and need a little or big push, **please ask for help!**

- Timebox your effort for each problem so that you don't spend your life on the problem sets. (See the problem solving tips guide for how to do this effectively.)
- Post in Discord. If you're following the timebox guide, you can post the exact statement that you produced after spending 20 minutes being stuck.
- Come to office hours or visit the CS Student Success Center.

Problem 1 (12 points)

*Hint:* See the lecture introducing asymptotic analysis for examples of proving and disproving that  $f(n) = O(g(n))$ .

- (a) (3 points) Prove that  $4n^3 + 2n^2 - n = O(n^3)$  by constructing  $c > 0, n_0 \geq 0 : \forall n \geq n_0 : 4n^3 + 2n^2 - n \leq c \cdot n^3$ .

**Grading Notes.** This rubric is straightforward: give a correct  $c$  and  $n_0$ .

- (b) (9 points) Prove that  $4n^3 + 2n^2 - n \neq O(n^2)$  by disproving  $\exists c > 0, n_0 \geq 0 : \forall n \geq n_0 : 4n^3 + 2n^2 - n > c \cdot n^2$ .

**Grading Notes.** While a detailed rubric cannot be provided in advance as it gives away the solution details, the following is a general idea of how points are distributed for this problem. We give partial credit where we can.

- (7) **Correctness.** If your proof is not correct, this is where you'll get docked. Make sure you model your proof off of examples of proofs that function  $f$  is not big O of function  $g$  from class or the textbook.
- (2) **Communication.** We need to see a mix of notation and intuition. If you skip too many steps at once, or we cannot follow your proof, or if your proof is overly wordy or confusing, this is where you'll get docked.

Problem 2 (15 points)

*Hint:* See the lecture introducing asymptotic analysis for examples of proving and disproving that  $f(n) = O(g(n))$ .

- (a) (2 points) Give two functions,  $f$  and  $g$ , such that  $f = O(n^2)$  and  $g = O(2^n)$  but  $f \neq O(g)$ .

**Grading Notes.** This rubric is straightforward: give a correct  $f$  and  $g$ .

- (b) (6 points) Prove that  $f = O(n^2)$  and  $g = O(n^3)$ .

- (c) (6 points) Prove that  $f \neq O(g(n))$ .

**Grading Notes.** For each of (b) and (c) above, here is the grading rubric:

- (5) **Correctness.** If your proof is not correct, this is where you'll get docked. Recall that a proof that  $f$  is big O of  $g$  involves specifying a  $c > 0, n_0 : \forall n \geq n_0 : f(n) \leq c \cdot g(n)$ , and a proof that  $f$  is not big O of  $g$  involves demonstrating that no such  $c, n_0$  can exist.
- (1) **Communication.** We need to see a mix of notation and intuition. If you skip too many steps at once, or we cannot follow your proof, or if your proof is overly wordy or confusing, this is where you'll get docked.

### Optional, Bonus Problem 3

We stated in class, without proof, that big O is transitive: if  $f(n) = O(g(n))$  and  $g(n) = O(h(n))$ , then  $f(n) = O(h(n))$ . Prove that this is true.

*Hint:* See the proof of Lemma 6.2, Asymptotic Equivalence of Max and Sum, for an example of proof of this type of property.

**This isn't graded, but you could think about....**

- Since this is a proof that a function is big O of another function, you must show that  $f(n)$  and  $h(n)$  conform to the definition of  $f(n) = O(h(n))$ ; that is, you must give a  $c > 0, n_0 \geq 0 : \forall n \geq n_0 : f(n) \leq c \cdot h(n)$ .
- These  $c, n_0$  must be correct.
- Regardless of how you formulate your proof, somewhere you'll need certain facts without which the proof wouldn't work. E.g., if it weren't true that the sum of two integers is an integer, would your proof fail? If so, then that is a fact I need to see stated somewhere.
- The order of these facts must make sense, so that you're not inferring something before you have all the facts to infer it. E.g., you cannot use the fact that the sum of two integers is integer if you don't already know that you have two integers to begin with.
- We need to see a mix of notation and intuition. If you skip too many steps at once, or we cannot follow your proof, or if your proof is overly wordy or confusing, that's not good.