

CMPS 6610 Problem Set 02

Name: _____

In this assignment we'll work on applying the methods we've learned to analyze recurrences, and also see their behavior in practice. As with previous assignments, some of your answers will go in `main.py`. Please add your written answers to `answers.md` which you can convert to a PDF using `convert.sh`. Alternatively, you may scan and upload written answers to a file named `answers.pdf`.

1. Prove that $\log n! \in \Theta(n \log n)$.

.
. .
. .
. .
. .

2. Derive asymptotic upper bounds for each recurrence below, using a method of your choice.

- $T(n) = 2T(n/6) + 1$.

.
. .
. .
. .

- $T(n) = 6T(n/4) + n$.

.
. .
. .
. .

- $T(n) = 7T(n/7) + n$.

.
. .
. .
. .

- $T(n) = 9T(n/4) + n^2$.

.
. .
. .
. .

- $T(n) = 4T(n/2) + n^3$.

.
. .

•
•

- $T(n) = 49T(n/25) + n^{3/2} \log n$.

•
•
•
•

- $T(n) = T(n - 1) + 2$.

•
•
•
•

- $T(n) = T(n - 1) + n^c$, with $c \geq 1$.

•
•
•
•

- $T(n) = T(\sqrt{n}) + 1$.

•
•
•
•

3. Suppose that for a given task you are choosing between the following three algorithms:

- Algorithm \mathcal{A} solves problems by dividing them into two subproblems of one fifth of the input size, recursively solving each subproblem, and then combining the solutions in quadratic time.
- Algorithm \mathcal{B} solves problems of size n by recursively one subproblems of size $n - 1$ and then combining the solutions in logarithmic time.
- Algorithm \mathcal{C} solves problems of size n by dividing them into a subproblems of size $n/3$ and a subproblem of size $2n/3$, recursively solving each subproblem, and then combining the solutions in $O(n^{1.1})$ time.

What is the work and span of these algorithms? For the span, just assume that it is the same as the work to combine solutions (i.e. the non-recursive quantity). Which algorithm would you choose? Why?

•
•
•

4. Suppose that for a given task you are choosing between the following three algorithms:

- Algorithm \mathcal{A} solves problems by dividing them into five subproblems of half the size, recursively solving each subproblem, and then combining the solutions in linear time.
- Algorithm \mathcal{B} solves problems of size n by recursively solving two subproblems of size $n - 1$ and then combining the solutions in constant time.
- Algorithm \mathcal{C} solves problems of size n by dividing them into nine subproblems of size $n/3$, recursively solving each subproblem, and then combining the solutions in $O(n^2)$ time.

What is the work and span of these algorithms? For the span, just assume that it is the same as the work to combine solutions (i.e., the non-recursive quantity). Which algorithm would you choose? Why?

5. In Module 2 we discussed two algorithms for integer multiplication. The first algorithm was simply a recapitulation of the “grade school” algorithm for integer multiplication, while the second was the Karatsuba-Ofman algorithm. For this problem, you will use the stub functions in `main.py` to implement these two algorithms for integer multiplication. Once you’ve correctly implemented them, test the empirical running times across a variety of inputs to test whether your code scales in the manner predicted by our analyses of the asymptotic work.

6. A “white hat” conducts hacking activities for the common good, while a “black hat” hacker does so for nefarious reasons. Let’s consider a computer security class with n students who are all either white hat or black hat hackers. You’re the instructor, and you don’t know who is a white hat or a black hat, but all of the student do.

Your goal is to identify the white hats and you’re allowed to ask a pair of students about one another. White hats will always tell the truth, but you cannot trust

a black hat's response. For a pair of students A and B then there are several possible outcomes:

A says	B says	Conclusion
B is a white hat	A is a white hat	both are good, or both are bad
B is a white hat	A is a black hat	at least one is bad
B is a black hat	A is a white hat	at least one is bad
B is a black hat	A is a black hat	at least one is bad

6a. Show that if more than $n/2$ students are black hats, you cannot determine which students are white hats based on a pairwise test. Note that you must assume the black hats are conspiring to fool you.

6b. Consider the problem of finding a single white hat, assuming strictly more than $n/2$ of the students are white hats. Show that $n/2$ pairwise interviews is enough to reduce the problem size by a constant fraction.

6c. Using the above show that all white hats can be identified using $\Theta(n)$ pairwise interviews.