

# (Take Home) Midterm Exam – Analysis of Algorithms

William David Shoaff

Spring 2022; Assigned: Wednesday, February 23, 9:00 A.M.; Due Saturday,  
February 26, 11:59 P.M.

Answer the following questions in the space provided
--

Name, major, and degree: \_\_\_\_\_

You are not to consult other people when answering these questions. You may consult sources (books, web pages, journals, *etc.* Cite your sources. Don't just copy and paste.

# 1 Fundamental Ideas and Definitions

Let  $f : \mathbb{N} \rightarrow \mathbb{R}$  and  $g : \mathbb{N} \rightarrow \mathbb{R}$  be *time complexity functions*.

1. (5 points) What does it mean to say “ $f(n)$  is big- $O$  of  $g(n)$ ”? Give a clear mathematical answer (You may translate to English, but be precise and use correct descriptive words).

2. (5 points) Prove that  $f(n) = n^2 + n + 2$  is  $O(n^4)$ . Provide *witnesses* that establish your proof.

## 2 Loops

### 2.1 Outcome

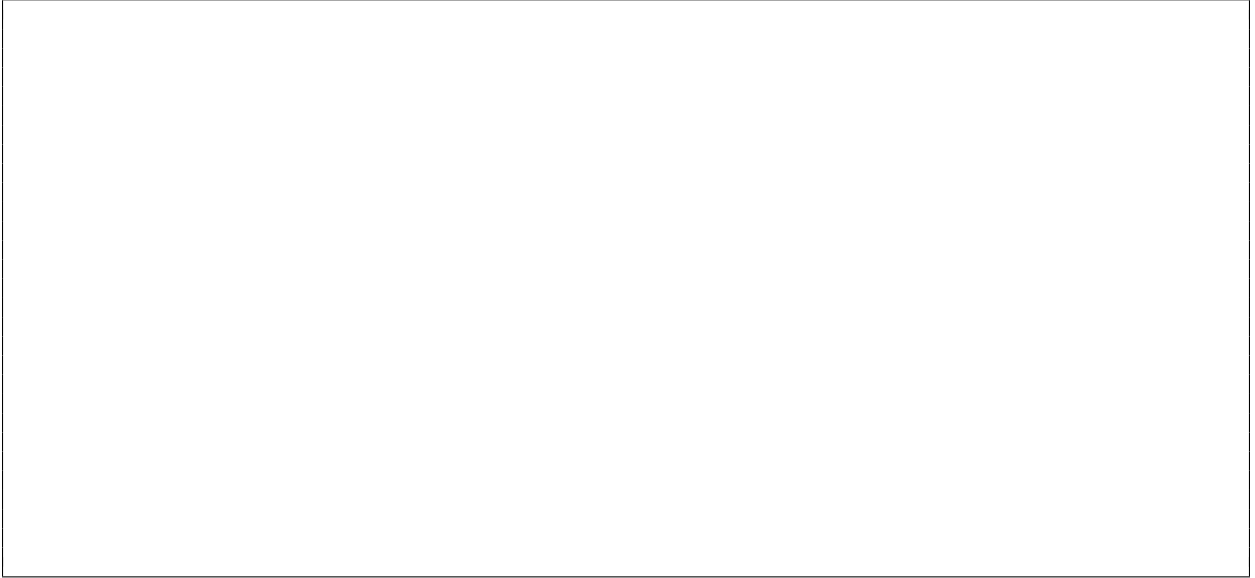
Students are able to use summations to calculate the time complexity of code snippets

3. (10 points) What is the value of sum after executing the code below and what is the time complexity of the code?

```
1  int snip(int m) {  
2      int sum = 0;  
3      for (int i = 1; i < m; i = power(2,i))  
4          {sum = sum + i;}  
5  }
```

4. (10 points) What will be the value of sum? after executing the code snippet below. Use summation notation to compute the time complexity of the code snippet below?

```
1      sum = 0;  
2      for (i = 0, i < n, i++) {  
3          for (j = n, j > 0, j--) {  
4              sum = sum + i * j;}}}
```



## 3 Recurrences

### 3.1 Outcome

Students are able to derive and solve recurrence relations given initial conditions, pseudo-code, or the name of a known algorithm.

5. (10 points) What initial conditions (or assumptions) must be **True** for *binary search* to execute correctly on an array of length  $n$ .

Also, what is the recurrence relation for binary search, and what is the solution of this recurrence?

## 4 Algorithms

### 4.1 Sorting Algorithms

Sorting and Searching are fundamental tasks in computing.

6. (10 points) Quicksort is a classic sorting algorithm. Describe it's best and worst case running time for sorting an array of size  $n$  by giving it's best case and worst case recurrences together with their solutions.

## 4.2 Searching Algorithms

Searching for a *pattern*  $p$  of length  $n$  in a *text* string  $t$  of length  $m$  is a common problem with many applications.

Several algorithms for this problem were presented in the notes and are also described online via simple searches.

To put this problem into context: Assume strings are over the DNA alphabet  $\{A, C, G, T\}$ .

Also, assume  $n = 5 \times 10^6$ , and  $m = 3.2 \times 10^9$ . That is, you are looking for long DNA sequence within the human genome.

Also, assume character-by-character comparisons can be completed in 1 second (unrealistically slow, but it simplifies the math). The point is to compute back-of-the-envelope calculations to get quick estimates of running time.

Note: One year is about  $32 \times 10^6$  seconds. Also, specifying units, as taught in physics courses is a good idea to ensure arithmetic operations are sensible. Each of these questions are for the worst-case of the given pattern matching algorithm.

7. (10 points) How long will brute-force pattern matching take?

8. (10 points) How long will Knuth-Morris-Pratt pattern matching take?

9. (10 points) How long will the Boyer-Moore pattern matching algorithm take?



### 4.3 Algorithms in General

### 4.4 Outcome

Students are aware that algorithms can be classified by paradigm.

10. (10 points) List three or more paradigms that can be used to classify algorithms.



11. (10 points) Some problems only require a “yes” or “no” answer. Other problems require a list of feasible answers, with one (or more) that is best by some measure.

What are the common names of these problem types?



Question	Points	Score
1	5	
2	5	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
9	10	
10	10	
11	10	
Total:	100	