## "I pledge my honor that I have abided by one stovens Honor System." Judion

CS 385, Homework 1b: Analysis of Algorithms

Name: Julia Welson

Date: 02/11/2020

Point values are assigned for each question.

Points earned: \_\_\_\_ / 100, = \_\_\_\_ %

1. Find an upper bound for  $f(n) = n^4 + 10n^2 + 5$ . Write your answer here:  $O(n^4)$  (4 points)

Prove your answer by giving values for the constants c and  $n_0$ . Choose the smallest integral value

possible for c. (4 points) 
$$n^4 + 10n^2 + 5 \le C \cdot n^4$$
  
 $n^4 + 10n^2 + 5 \le 2n^4$   $n^4 + 10n^2 + 5 \le 2n^4$ 

 $n^{4}+10n^{2}+5 \leftarrow 2n^{4}$   $n^{4}+10n^{2}+5 \leftarrow 2n^{4}$  =2  $45 \leftarrow 16X$  =3  $95 \leftarrow 8+X$  =4  $165 \leftarrow 256$ 

2. Find an asymptotically tight bound for  $f(n) = 3n^3 - 2n$ . Write your answer here:  $\Theta(n^3)$  (4) C: (n3) & 3n3-2n & Cz. (n3)

Prove your answer by giving values for the constants  $c_1$ ,  $c_2$ , and  $n_0$ . Choose the tightest integral

values possible for 
$$c_1$$
 and  $c_2$ . (6 points) let  $3n^3 - 2n \le c_2 \cdot n^3$ 

$$c_1 = 2$$

$$2n^3 \le 3n^3 - 2n$$

$$n = 1 2 \le 1 \times 1$$

$$n = 2 1 \le 20$$

$$3 \cdot 183n = 460(n^2)?$$
 Circle your answer, yes  $(n^3)$ ? points)

3. Is  $3n-4 \in \Omega(n^2)$ ? Circle your answer: yes (no.) 2 points)  $n_0=2$ 

If yes, prove your answer by giving values for the constants c and  $n_0$ . Choose the smallest integral

value possible for c. If no, derive a contradiction. (4 points)

n2 to be lower bound 12 4 3 n - 4 i = n= Z = 4 = 2 x n= 14 = 16 = 1 = 14 = 16 = 1

4. Write the following asymptotic efficiency classes in increasing order of magnitude.  $O(n^2)$ ,  $O(2^n)$ , O(1),  $O(n \lg n)$ , O(n), O(n),  $O(n^3)$ ,  $O(\lg n)$ ,  $O(n^n)$ ,  $O(n^2 \lg n)$  (2 points each)

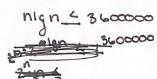
O(1) O(1gn) O(n) O(n1gn) O(n2) O(n2lgn) O(n3) O(2n), O(n1)

- 5. Determine the largest size n of a problem that can be solved in time t, assuming that the algorithm 15 = 1000 ms takes f(n) milliseconds. n must be an integer. (2 points each)
  - a. f(n) = n, t = 1 second

1000

n millise conds

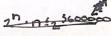
b.  $f(n) = n \lg n, t = 1 \text{ hour } 204094$ 









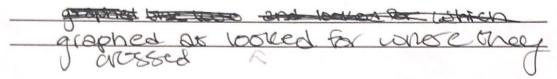


c.  $f(n) = n^2$ , t = 1 hour  $\frac{1897}{4}$   $n^2 = 3600000$ 

e. f(n)=n!, t=1 minute | 1 min = 60000 ms

N: = n = (n-1)! 120 = 3 = 304018V

6. Suppose we are comparing two sorting algorithms and that for all inputs of size n the first algorithm runs in  $4n^3$  seconds, while the second algorithm runs in  $64n \log n$  seconds. For which integral values of n does the first algorithm beat the second algorithm? 24n 24n 26 (4 points) Explain how you got your answer or paste code that solves the problem (2 point):



7. Give the complexity of the following methods. Choose the most appropriate notation from among O,  $\Theta$ , and  $\Omega$ . (8 points each)

int function1(int n) { int count =  $\theta$ ; for (int i = n / 2; i <= n; i++) { for (int j = 1; j <= n; j \*= 2) { count++; return count; Answer: O(n 1gn) int function2(int n) { int count =  $\theta$ ; for (int i = 1; i \* i \* i <= n; i++) { count++; return count; (cuberoot) Answer:  $\Theta$  ( $\sqrt{n}$ ) int function3(int n) { int count = 0; for (int i = 1; i <= n; i++) { for (int j = 1; j <= n; j++) { for (int k = 1; k <= n; k++) { count++;

```
return count;
Answer: 0(n3)
int function4(int n) {
     int count = 0;
     for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j++) {
             count++;
             break;
     return count;
Answer: 6(n)
int function5(int n) {
     int count = 0;
     for (int i = 1; i <= n; i++) {
         count++;
    for (int j = 1; j <= n; j++) {
        count++;
    return count;
Answer: \Theta(n)
```