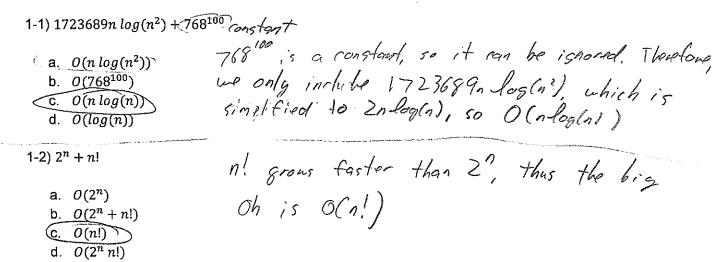
Oslan Mackey

CS251 - Homework 1: Algorithm Analysis

Out: January 15, 2016 @ 9:00 pm Due: January 22, 2016 @ 9:00 pm

Important: Each question has only one correct answer. Additionally, you must provide an explanation on each question. Any choice without an explanation, even though it is correct, will be graded with 0 points.

1) Select the tightest big-Oh expression for the following expressions:



2) Select the tightest big-Oh expression for the following pseudo codes:

2-1) int sum = 0; for(i = 0; i < n; i++) \cap for(j = 0; j < i; j++) \cap for(k = 0; k < j; k++) \cap sum+=sum;

b.
$$O(n^2 log(n))$$

c. $O(i^n j^n)$
d. $O(n^3 log(n))$

Sunt=sum is exercised at most non.n times,

2-2) for(inti=1; i <= n; i=2) los(a) - since i is multipliered by 2 each time, this is executed log(4) time. for(int j = 1; j <= n^2 ; j++) n^2 prod*=prod: 6(1) Upper bound is no, so prost no times a. $O(n^3)$ O(12 log(n)) is the max # of times exacted. \bigcirc b. $O(n^2 \log(n))$ c. $O(prod^n j^n)$ d. $O(prod^3 log(n))$ 2-3) for(i = 0; i < n; i++) for (j=1;j< i;j*=2) log(n)-1 - j starts @ land is incremented by $\times 2$, O(k) work where k is a constant 50 log(a)-1 a. $O(n^3)$ $b O(n \log(n))$ n(lag(1)-1) c. $O(n \log(n) j^k \log(n))$ n-log(0) - 1 [= 0(n-log(n)) d. $O(n^3 \log(n))$ 3) Find the big- Θ expression for the following big-Oh and big- Ω expressions: 3n2log(n)-n **3-1)** $O(3n^2 \log(n) - n)$ <3n2 log(r)+n2 log 2 (a. $\theta(n^2 \log(n))$) E 4n Tlog(n) b. $\Theta(n)$ c. $\Theta(n^2)$ < On log(n) d. $\Theta(n \log(n))$ Eiz= n(n+1) (2n+1) = Summation equation **3-2)** $\Omega(\sum_{i=0}^{n} i^2)$ a. $\Theta(n)$ = (n34N(ZN1) b. $\Theta(n \log(n))$ C. $\Theta(n^2)$ $d. \Theta(n^3)$ $\frac{2n^{3}+n^{2}+2n^{2}+n}{6}$ $= 2n^{3}+3n^{2}+n$

a. $log(n)$ b. n c. $n log(n)$ d. 1	By using the binary search method, we would return a worst case (cenario of O(log(n)),
•	50 the long-est time is 2,670,286,176
Approach A) If you a much from day to day website popularity son that day). This a $\frac{\text{Approach B}}{O(n^2)}$ If you if $O(n^2)$ sorting algorithms.	form a daily re-sorting of the most visited student websites from CS251. You ider the following approaches (in all of them n is the number of students): assume that the popularity of the websites among students do not change ay, you can implement an $O(n)$ average running time algorithm to keep the corted in daily manner (n is the number of students who search the website algorithm costs 100 time units per step to perform. Os $\frac{1}{2} \frac{1}{2} $
Approach C) Assum	In that day). This algorithm costs 5 time units per step to perform. $ (cs+=5)^2 $ in the you are provided with an $O(n,log(n))$ sorting algorithm but we do not a salgorithm per unit. However, we know that 100 students are registered to $ (cs+=100 \log 100) $
Note: Base of the lo	ogarithm is 10 for all the calculations.
6-1) How many stud <i>Oh</i> time complexity	dents should register to CS251 so Approaches A and B have the same <i>big</i> -?

A?

6-2) What should be the cost per time unit for Approach C so it behaves the same as Approach Α?

106(100) - 100 x log (100) to equation a. 1 100 = x log (100) X= Tag(100) (b. 50) c. 100 d. 105 x=50

Submit Instructions:

The homework must be turned in by the due date and time using the turnin command. Follow the next steps:

- 1. Login to data.cs.purdue.edu (you can use the labs or a ssh remote connection).
- 2. Make a directory named with your username and copy your solution (in pdf format) there.
- 3. Go to the upper level directory and execute the following command:

turnin -c cs251 -p hw1 your_username

(Important: previous submissions are overwritten with the new ones. Your last submission will be the official and therefore graded).

4. Verify what you have turned in by typing turnin -v -c cs251 -p hw1 (Important: Do not forget the -v flag, otherwise your submission would be replaced with an empty one).