Review Test Submission: Midterm 2

User	Yaokun Wu
Course	CS 5333.001 - Discrete Structures - F20
Test	Midterm 2
Started	11/2/20 8:33 AM
Submitted	11/2/20 10:02 AM
Status	Completed
Attempt Score	75 out of 100 points
Time Elapsed	1 hour, 29 minutes out of 1 hour and 30 minutes
Instructions	The exam will end 90 minutes after you begin. As mentioned in the email instructions, all answers must be typed directly into the exam. You are only allowed to use the slides of the course and notes that you may have taken during the videos of the course. Other material is strictly prohibited. Keep your camera and microphone on at all times. Keep your speakers on also at all times. If you need to reach me send me email. Best wishes, JC
Results Displayed	All Answers, Submitted Answers, Correct Answers, Feedback, Incorrectly Answered Questions

Question 1

2 out of 2 points

I will only use the slides of the course and my notes from the lectures to answer this exam. I will not interact with other students or other persons in any way during the exam.

Selected Answer: True

Answers: True

False

Question 2

14 out of 14 points

(Growth Functions)
Consider the following two functions:

$f(n) = 6n^2$

$g(n) = n^2\log(n)$

(a) Is $f(n) \in O(n)$? (big-Oh) Answer YES or NO, and then give an explanation as to why.

(b) Is $f(n) \in \Omega(n)$? (big-Omega) Answer YES or NO, and then give an explanation as to why.

(c) Is $f(n) \in \theta(n)$? (big-Theta) Answer YES or NO, and then give an explanation as to why.

Selected Answer:

a).
for f(n) : No, we want to find : Exist k, Exist C, ForAll n [n > k -> |6n^2| <= C*|n|] , after simplification :|6n| <= C, but no matter what C value is, |6n| will exceed C at some point.
for g(n) : No, we want to find : Exist k, Exist C, ForAll n [n > k -> |n^2*log(n)| <= C*|n|], after simplification: |n*log(n)| <= C, but no matter what C value is, |n*log(n)| will exceed C at some point.
b).
for f(n) : Yes, we want to find : Exist k, Exist C, ForAll n [n > k -> |6n^2| >= C*|n|]. So, let k = 1 and C = 1. |6n| >= C will hold as long as n > k = 1.
for g(n): Yes, we want to find : Exist k, Exist C, ForAll n [n > k -> |n^2*log(n)| >= C*|n|]. So, let k = 10 and C = 1, |n*log(n)| >= C will hold as long as n > k = 10.
c).
for f(n) : No, because the upper bound is not O(n).
for g(n): No, because the upper bound is not O(n).

Correct Answer: [None]

Response Feedback: [None Given]

Question 3

14 out of 14 points

(Growth Functions Theory)

Prove whether or not each of the following statements are true. For those that you believe are false, prove this by giving a counterexample (i.e., particular functions for f(n) and g(n) for which the statements are not true). For those that you believe to be true, use the formal definitions of big-O, big-Omega, and big-Theta to prove it (only the definitions, do not use a theorem proven in class without proving it yourself). In all problems, you are given that for all n, f(n) >= 0 and g(n) >= 0.

(a) If $f(n) \in \Omega(g(n))$ then $g(n) \in O(f(n))$

(b) If $f(n) \in O(g(n))$ then $g(n) \in O(f(n))$

(c) If $f(n) \in \theta(g(n))$ and $g(n) \in \theta(h(n))$ then $f(n) \in \theta(h(n))$

Selected Answer:

a). This statement is true.
By definition of big omega: Exist k, Exist C, ForAll n [n > k -> |f(n)| >= C*|g(n)|], assume here k = k1 and C=C1,
which means that: ForAll n [n > k -> |g(n)| <= 1/C*|f(n)|] at k = k1 and C = 1/C1, so g belongs to O(f(n)).
b). This statement is false.
Assume g(n) = n^2 and f(n) = n, so that f(n) belongs to O(g(n)) but g(n) doesn't belongs to O(f(n)).
c). This statement is true.
By definition of big theta:
Exist k, Exist C, ForAll n [n > k -> |f(n)| <= C*|g(n)|], assume here k = k1 and C=C1 and
Exist k, Exist C, ForAll n [n > k -> |f(n)| >= C*|g(n)|], assume here k = k2 and C=C2
Exist k, Exist C, ForAll n [n > k -> |g(n)| <= C*|h(n)|], assume here k = k3 and C=C3 and

Exist k, Exist C, ForAll n [n > k -> |g(n)| >= C*|h(n)|], assume here k = k4 and C=C4

-> C2|g(n)| <= |f(n)| <= C1|g(n)| and C4|h(n)| <= |g(n)| <= C3|h(n)|

Let k' = max(k1, k2, k3, k4).

1/C2|f(n)| >= |g(n)| >= C4|h(n)| -> |f(n)| >= C2*C4 |h(n)| -> f(n) belongs to BigOmega(h(n)) where k = k' and C = C2*C4

1/C1|f(n)| <= |g(n)| <= C3|h(n)| -> |f(n)| <= C1*C3 |h(n)| -> f(n) belongs to O(h(n)) where k = k' and C = C1*C3

Since f(n) is both BigOmega(h(n)) and O(h(n)), then f(n) belongs to BigTheta(h(n)).

Correct Answer: [None]

Response Feedback: [None Given]

Question 4

10 out of 14 points

(Algorithm Complexity)

In the algorithms below, show the "best" big-Oh notation to describe the (worst-case) time complexity of the given algorithm.

Give a brief (one sentence or two) explanation justifying your choice.

a) A binary search of n numbers

b) A search to find the smallest number in a list of unordered numbers

c) An algorithm that prints to the screen all the ways in which the numbers 1, 2, 3, . . . n, can be written in a row (assuming printing a number to the screen takes constant time)

d) An algorithm that prints all the bit strings of length n except those bit strings that contain exactly three zeroes (assuming that printing one bit takes constant time).

Selected Answer:

a). O(nlogn) because search one number requires log(n) time and n numbers requires nlogn time.
b). O(n) because it requires the traversal of the whole list of size n.
c). O(n) because it will print n numbers and each print cost constant time.
d. O(n^2) because the worst case is printing all the strings and for each string, it takes n time, so a total of n*n = n^2.

Correct Answer: [None]

Response Feedback: c -3

d -1

Question 5

11 out of 14 points

(Induction)

Consider proving using induction the following predicate P(n), where n is a non-negative integer.

$$\sum_{j=0}^{n}(j+1)=(n+1)(n+2)/2$$

a) Prove the basis

b) Show the inductive step (don't prove it yet just show it)

c) Prove the inductive case.

Selected Answer:
a). n = 0, 1 == (1)*(2) / 2, which is true.
b). Assume P(k) is true where k >= 1, which means that P(k) = (k + 1)(k + 2)/2
c). P(k + 1) = 1 + 2 +...+ k + 2 = P(k) + k + 2 = (k + 1)(k + 2)/2 + k + 2 = ((k+1) + 1)((k+1) + 2)/2 as desired.

Correct Answer: [None]

Response Feedback: b -2 that is the ind hyp, I asked for the ind step

c -1 P(K) is the predicate not the sum

Question 6

14 out of 14 points

(Permutations and Combinations)

a) How many ways are there to arrange the letters of the word stubbornness? Show me the result using factorials (no need to calculate the actual number). Show your intermediate work.

b) How may ways are there to choose 15 cookies if there are 4 varieties of cookies? Show me the result using factorials (no need to calculate the actual number). Show your intermediate work.

Selected Answer:
a). Length = 12. s: 3, t:1, u:1, b:2, o:1, r:1, n:2, e:1. so answer = n!/(n1!*n2!*..*nn!) = 12!/(3!*2!*2!)
b). | | |, C(n + r - 1, r) = C(4 + 15 - 1, 15) = C(18, 3) = 18!/(15!*3!)

Correct Answer: [None]

Response Feedback: [None Given]

Question 7

3 out of 14 points

(Pigeonhole Principle)

Assume computer passwords are 5, 6, or 7 characters long. Each character can be one of the following: a lower case letter, an upper case letter, or a digit.

Assume each password must have at least one lower case letter, at least one upper case letter, and least THREE digits.

a) Let S be the set of passwords. What is the minimum cardinality of S that will guarantee that at least two passwords in S have the same length AND the same number of lower case letters?

b) Let S be the set of passwords. What is the minimum cardinality of S to guarantee that at least two passwords in S have the same length, the same number of lower case letters, AND the same number of upper case letters?

Selected Answer: Count of length 5: C(5,3) * 10^3 * C(2,1)* 26 * 26

Correct Answer: [None]

Response Feedback: [None Given]

Question 8

7 out of 14 points

(Advanced Counting)

For the questions below, give me the result using factorials (no need to compute the actual number). Show your intermediate work! Assume we have 19 people in 5 different rooms.

a) How many ways are there to assign people to the different rooms if we don't care about the differences between each person? (i.e. we don't care about names or any other characteristic of a person)

b) How many ways are there to assign people to the different rooms if we do care about the differences between each person (i.e., each person has a distinct name and we do care about names)

Selected Answer: a). | | | |, C(n + r - 1, r) = C(19 + 5 - 1, 19) = C(23, 4) = 23!/(19!*4!)
b). After assigning the person without difference, we make permutation of them to represent the difference so that answer is C(23,19)*P(19,19) = 23!/4!

Correct Answer: [None]

Response Feedback: b Xx

Thursday, December 3, 2020 4:35:52 PM CST

← OK