Quiz #1
Introduction to Algorithms/Algorithms 1
600.363/463

March 8th, 9:00-10:15am

Ethics Statement

I agree to complete this exam without unauthorized assistance from any person, materials, or device.

Signature Date

1 Problem 1 (10 points: each subproblem is 2 points)

For each statement below state if it is true or false.

1. Let f, g be two positive functions. If $f(n) = \Theta(n)$ and $g(n) = \Theta(n)$ then $|f(n/3) - g(n)| = \Theta(n)$.

   true false

2. Let f, g be two positive functions. If $f(n) = O(g(n))$ and $h(n) = \Omega(g(n))$ then $f(n) = O(h(n))$.

   true false

3. $\log_3(n) = \Theta(\log_5(n))$.

   true false

4. $n^2|\sin n| = \Theta(n^2)$.

   true false

5. $n^2 + (\log(n))^{\frac{n}{10}} = \omega(n^3)$

   true false

2 Problem 2 (20 points; each subproblem is 10 points)

Give asymptotic upper bounds for the following recurrences. You can use Master theorem, when it is applicable. Assume that $T(1) = 1$.

1. $T(n) = 9T(n/3) + \sqrt{n}\log(n)$

2. $T(n) = 2T(n/4) + n$

3 Problem 3 (60 points)

Suppose you are given an array A with n sorted numbers that has been circularly shifted k positions to the right. For example consider sorted array $\{5, 15, 27, 29, 32, 45\}$, then $\{32, 45, 5, 15, 27, 29\}$ is a sorted array that has been circularly shifted $k = 2$ positions, while $\{27, 29, 32, 45, 5, 15\}$ has been shifted $k = 4$ positions.
Suppose you do not know what k is. Give a worst case $O(\log n)$ time algorithm to find the largest number in A. Prove correctness and provide running time analysis.

30 points will be given if (1) your algorithm works correctly, (2) your algorithm solves the problem by using the time worse than $O(\log n)$ time, (3) your analysis is correct and (4) your explanations and proofs are clear and with enough details. If you cannot prove your claims formally, give your best intuition.

The full credit will be given if (1) your algorithm works correctly, (2) your algorithm solves the problem in $O(\log n)$ time, (3) your analysis is correct and (4) your explanations and proofs are clear and with enough details. If you cannot prove your claims formally, give your best intuition.

4 Problem 4 (60 points)

Let A be an array of integers from the range $[1, \ldots, n^2]$, where the same number can appear multiple times. The size of the array is n. Design an efficient algorithm that finds the most frequent item in A. For example, consider $n = 12$ and $A = [4, 5, 5, 3, 1, 7, 4, 3, 4, 4, 3, 4]$, then 4 is the most frequent item in A. If there are two or more most frequent items like in the case $[3, 2, 1, 2, 3, 1, 2, 1, 3, 4, 4, 4]$ your algorithm can output any of them. Prove the correctness of your algorithm and provide running time analysis.

30 points will be given if (1) your algorithm works correctly, (2) your algorithm solves the problem in $O(n^2)$ time, (3) your analysis is correct and (4) your explanations and proofs are clear and with enough details. If you cannot prove your claims formally, give your best intuition.

45 points will be given if (1) your algorithm works correctly, (2) your algorithm solves the problem in $O(n \log n)$ time, (3) your analysis is correct and (4) your explanations and proofs are clear and with enough details. If you cannot prove your claims formally, give your best intuition.

The full credit will be given if (1) your algorithm works correctly, (2) your algorithm solves the problem in $O(n)$ time, (3) your analysis is correct and (4) your explanations and proofs are clear and with enough details. If you cannot prove your claims formally, give your best intuition.

Note: we have not covered hashing in the class, thus you can not use it in this problem.