CS4231: Analysis of Algorithms, I

Midterm Exam, Thursday October 25, 2018

This exam contains 6 problems, some of them composed of several parts. There are 100 points in all, and you have 75 minutes. Do not spend too much time on any problem. Read them all through first and attack them in the order that allows you to make the most progress.

The exam is closed book and closed notes. You may use one 8.5”x11” crib sheet (both sides).

You can use any algorithm that we covered in class or the homeworks by simply referring to it and specifying the input. You do not need to repeat the algorithms.

Write your solutions in the space provided (and only there).
You can use the blank pages at the end of this booklet for scratch space.
Be *clear, precise and succinct* in your answers. You will be graded not only on the correctness of your answers, but also on the clarity with which you express them.
Be neat.

Good luck!

NAME (Last, First): ______________________________________

UNI: __________________________

Problem	Max Points	Points
1	18	
2	16	
3	10	
4	18	
5	18	
6	20	
Total	100	

Problem 1 [18 points, 6 points per part]

Give asymptotic solutions $T(n) = \Theta(f(n))$ for the following recurrences. Assume that $T(n)$ is constant for sufficiently small n. State in each case the method that you used (eg. Master method, etc.). No justification is needed. For example, you can say, "$T(n) = \Theta(n)$, I used the Master method." No more explanation is required.

1. $T(n) = 3T(n/2) + n$

2. $T(n) = 2T(n-1) + 1$

3. $T(n) = 2T(n/3) + T(n/4) + n$

Problem 2 [16 points, 4 points per part]

For each pair of functions *f*, *g* below, determine whether *f*=o(*g*), *f*=Θ(*g*), or *f*=ω(*g*) and circle the correct answer. No justification is needed. All logarithms are with base 2.

1. $f(n) = 4n^4 + 2n^3 + 10$, $g(n) = (n^2 - 1)^2$ *f*=o(*g*) *f*=Θ(*g*) *f*=ω(*g*)

2. $f(n) = 2^{n+5}$, $g(n) = 2^{2n}$ *f*=o(*g*) *f*=Θ(*g*) *f*=ω(*g*)

3. $f(n) = n\sqrt{n}$, $g(n) = n(\log n)^4$ *f*=o(*g*) *f*=Θ(*g*) *f*=ω(*g*)

4. $f(n) = n^4$, $g(n) = 2^{3\log n}$ *f*=o(*g*) *f*=Θ(*g*) *f*=ω(*g*)

Problem 3 [10 points]
Consider the following game. There are n cards with the numbers $1,\dots,n$ written on them, where each of the numbers $1,\dots,n$ is written on exactly one of the cards. The cards are face down on the table, so we do not see their numbers. We turn over the cards one by one. Before turning each card, you guess the number written on the card; you gain 1 point if you guess correctly and 0 if incorrectly. A random strategy is to guess in each step uniformly at random a number that has not appeared so far in the previous cards.

1. (2 points) What is the probability that the guess in the i-th step is correct?

2. (3 points) Let $T(n)$ be the expected total number of points that you gain using the random strategy. Which of the following functions describes the asymptotic growth rate of $T(n)$? Circle the correct answer.

$\Theta(1)$ $\Theta(\log n)$ $\Theta(\sqrt{n})$ $\Theta(n)$ none of these

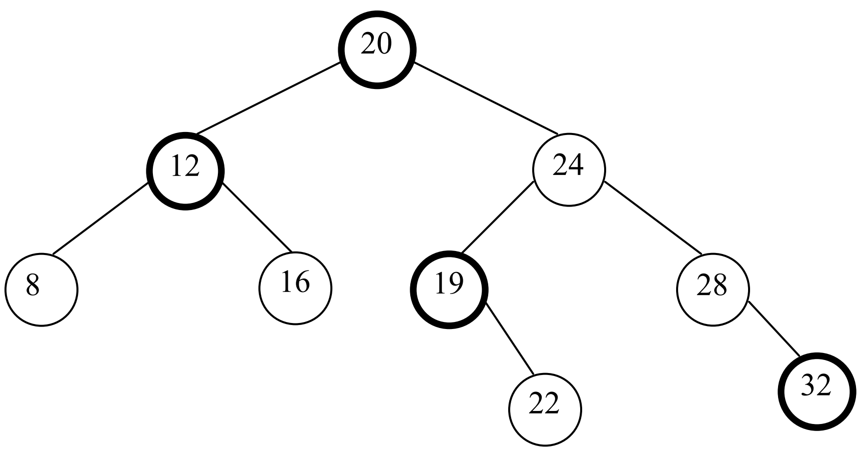
3. (5 points) Justify your answer of part 2.

Problem 4 [18 points; 6 points per part]

Answer the following questions. In the True-False questions, circle True or False and justify your answer by either providing an algorithm (or pointing to a known algorithm), or by providing an argument why the task in the claim is impossible. Your justifications are *more important* than the True/False designations, and must be ***succinct, precise and convincing***.
All the parts can be adequately answered and justified in a few lines.

All the times below refer to worst-case time.

1. Consider the following tree where the dark nodes are black and the light nodes are red. Give three reasons why this is not a legal red-black tree.



Reason 1:

Reason 2:

Reason 3:

2. *Claim*: Given an (unsorted) array A with n integers in the range 1 to n^2, we can determine in $O(n)$ worst-case time whether there exist two indices i,j such that $A[i]+A[j]= n^2$.

True False

3. The new congress will simplify the tax code as follows.
There will be only 3 tax brackets:

- the top 10% of people with highest income will pay 35% tax
- the next 20% of people in order of income will pay 20% tax
- the bottom 70% of people will pay 5% tax

We are given an array $A[1 \ldots n]$ with the records of all the people and their incomes; the array is sorted by name. Assume that no two people have the same income.

Claim: We can compute the taxes of all the people in O(n) worst-case time.

True False

Problem 5 [18 points]

Consider the following algorithm.

Input: Array $A[1\ldots n]$ of numbers
Output: Array $B[2\ldots n]$

```
for i=2 to n do
  {  B[i] = ∞;
     for j= 1 to i-1 do
        if ( 0 ≤  A[i]-A[j] < B[i] ) then B[i] = A[i]-A[j]
  }
return B
```

1. [6 points]
Give the time complexity of this algorithm in $\Theta(.)$ form. Justify your answer.

2. [12 points]
Give a more efficient algorithm that computes the same output. You do not have to give pseudocode; clear, precise description in English suffices. State explicitly the (worst-case) running time of your algorithm, and justify its correctness and the running time. (*Hint:* Use a suitable data structure.)

Problem 6. [20 points]

We have n coins, all of which have the same weight except for one that is fake and weighs less. We have a scale which we can use to compare the weights of any two (disjoint) subsets of coins: if we place a subset A of coins on one side and subset B on the other side, the scale tells us whether the two sides have the same weight, and if not, it tells us which side is heavier. Note that every weighing has *three* possible outcomes: A is heavier, or B is heavier, or A and B have equal weight. Each use of the scale counts as one step. We would like to identify the fake coin in the minimum number of steps.

1. [12 points] Give an algorithm that identifies the fake coin using as few steps as you can. Give the exact number of steps (not just an asymptotic expression). Justify the correctness of your algorithm and the running time.

2. [8 points] Show a lower bound on the number of steps required to identify the fake coin. The lower bound should be again in the form of an exact expression (not just asymptotic).