Midterm Exam

Instructions: This is a closed book, closed notes exam. Do all **3** problems. Communicate your ideas ***clearly*** and ***succinctly***. Show all work. Good luck!

Your name:

Your e-mail address (neatly!):

On problem	you got	out of
1		35
2		35
3		35
Σ		105

1 Short answers [35 points]

1.1. Carefully define what it means for a family of hash functions $\mathcal{H} = \{h : U \to \{0, \ldots, m-1\}\}$ to be ε-***almost universal***.

1.2. Any comparison-based sorting algorithm needs at least this many comparisons, in the worst case, to sort 6 numbers.

1.3. Recall the QUICKSORT-like **selection algorithm** that we described in class: to find the i-th smallest element in the array $A[1..n]$, choose a random "pivot" x in A, rearrange the elements of A so that elements $A[1..p-1]$ are all less than or equal to x, $A[p] = x$, and the elements of $A[p+1..n]$ are all greater than or equal to x. Now recurse in the appropriate piece of A.

Suppose, instead of choosing a random pivot, your algorithm, using $5n$ comparisons, manages to find a pivot x of rank p where $\frac{n}{3} \leq p \leq \frac{2n}{3}$. Write a recurrence relation which specifies the worst-case number of comparisons, $T(n)$, of this version of the selection algorithm.

1.4. Solve the following recurrence relation, giving your answer in Θ-notation. Assume $T(n)$ is constant for sufficiently small n. Show your work or otherwise justify your answer. (If you've seen the recurrence before then saying where, and what is the solution, is an adequate justification).

$$T(n) = 9T(n/3) + n^2 + n\lg n$$

1.5. Same instruction as in 1.4:

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

1.6. Same instruction as in 1.4:

$$T(n) = T(n/2) + \lg n$$

2 Dynamic Programming [35 points]

It the Kingdom of Konfusion there are coins of denominations $1 = C_1 < C_2 < \cdots < C_K$. You have been hired to write a program which will calculate, for any integer $N \geq 1$, the number of ways to make N cents using coins of the given denominations.

For example, if the coins have denominations $(C_1, C_2, C_3) = (1, 5, 10)$ ("pennies", "nickels", and "dimes"), and you want to make 15 cents in change, then there are **6** ways to do this: (1) a dime and a nickel; (2) a dime and five pennies; (3) three nickels; (4) two nickels and five pennies; (5) a nickel and ten pennies; or (6) fifteen pennies.

More generally, let $A[n, k]$ be defined as the number of different ways to make n cents using coins of denominations $C_1, C_2, \ldots C_k$.

A. Write a recurrence relation for $A[n, k]$. For full credit you will need to correctly specify the base case as well as the inductive part.

B. Write a program (in pseudocode) to calculate $A[N, K]$.

C. What is the running time of your program? (In Θ-notation).

3 Justified True/False [35 points]

Put an **X** through the **correct** box, and give a brief (but convincing) justification.

3.1. Let $H(n) = 1 + \frac{1}{2} + \frac{1}{3} + \cdots \frac{1}{n}$. Then $H(n) \in \Theta(n \lg n)$. **True** **False**

3.2. Recall the "proposal algorithm" for solving the stable-marriage problem ("man proposes, woman disposes"). Suppose that there is a man A and a woman a such that A's favorite woman is a, and a's favorite man is A. Then, no matter what choices are made in running the proposal algorithm, A and a will eventually end up married. **True** **False**

3.3. For any pair of functions f and g, both from the positive integers to the positive reals, either $f \in O(g)$ or $g \in O(f)$. **True** **False**

3.4. Our algorithm to find the closest pair of points in the plane took $\Theta(n^2)$ time, in the worst case. **True** **False**

3.5. There is an algorithm to sort n integers between 0 and $10n$ in $O(n)$ time. **True** **False**
