Midterm Exam

Instructions: This is a closed book, closed notes exam. Do all **3** problems. Communicate your ideas ***clearly*** and ***succinctly***. Show all work. There are two blank sheets at the end of the exam in case you need them.

NAME:

On problem	you got	out of
1		50
2		25
3		25
Σ		100

1 Short answers [50 points]

1.1. *(5 points)* Carefully define what it means for a hash function $h : U \to [0..m-1]$ to be ***perfect*** on $S \subseteq U$.

1.2. *(5 points)* Carefully define what it means for a family of hash functions $\mathcal{H} = \{h : U \to [0..m-1]\}$ to be ***universal***.

1.3. *(5 points)* Define the ***amortized time*** for an operation `op` on a data structure DS.

1.4. *(6 points)* Recall that a sorting algorithm is said to be ***in place*** if it requires only a constant amount of additional space to sort a list of numbers presented in an array. Is QUICKSORT in-place? Justify your answer.

1.5. *(10 points)* You are given 5 distinct numbers to sort. Describe an algorithm which sorts them using at most 6 comparisons, or argue that no such (comparison-based) algorithm exists.

1.6. *(6 points)* Solve the following recurrence relations, finding a closed form expression $T(n) = \Theta(\cdots)$. Assume $T(n)$ is constant for sufficiently small n. Show your work or otherwise justify your answer.

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

1.7. *(6 points)* Same instruction as in part (F):

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

1.8 *(7 points)* Suppose you do a SEARCH on the element with key 20 in the following ***splay tree.*** Draw the tree that results when you finish all splays produced by this search. Show any intermediate steps. *(Be careful — no partial credit.)*

2 Dynamic Programming [25 points]

There are n trading posts along a river, $n \to n-1 \to \cdots \to 2 \to 1$, numbered such that trading post n is furthest upstream and trading post 1 is furthest downstream. At any of the trading posts you can rent a canoe, which may be returned at any other trading post downstream. (It is impossible to paddle against the current and go upstream.) For each possible departure point i and each possible arrival point j, $i > j$ the cost $c[i,j]$ of renting a canoe to go directly from i to j is known. Different companies run the rentals and, because of this, the cost of renting from i to j can be higher than the total cost of a series of shorter trips. In that case you may want to make an i-to-j trip by renting a canoe at i and returning it at some intermediate trading post; where you rent a different canoe and return it at some other intermediate trading post; and so forth, until you reach j. There is no additional cost for changing canoes in this way.

2.1. *(8 points)* Let $C_{i,j}$ be the cost of a least-expensive routing from i to j. Write an expression for $C_{i,j}$ which is useful for developing a dynamic programming solution to this problem.

2.2. *(13 points)* Write clear and correct pseudo-code for a (reasonably efficient) program which, on input n, prints out a cheapest routing from trading post n to trading post 1. *(Suggestion: work this out on scratch paper then copy neatly into the space below.)*

2.3. *(4 points)* State the time complexity of your program (as $\Theta(\cdot)$) as a function of n.

3 Algorithm design [25 points]

Suppose you are given a set of n real numbers, $X = \{x_1, x_2, \ldots, x_n\}$. Your job is to find a ***minimum-cardinality*** set of unit intervals $I = \{I_1, I_2, \ldots, I_m\}$ which "cover" all the points in X — that is, each $I_i = [a_i, b_i]$ is a closed interval of length 1 ($b_i - a_i = 1$); each x_i lies in at least one interval I_j; and there is no small number of unit intervals I' for which each x_i is in at least one of the intervals $I'_j \in I'$.

3.1. *(11 points)* Give an algorithm which finds such a minimum-cardinality set of intervals I and runs in worst-case $O(n \lg n)$ time.

3.2 *(14 points)* Prove that your algorithm works.