Midterm Exam, Section 2 (9:00)

NAME:

Instructions: This is an open book, open notes exam. Communicate your ideas ***clearly*** and ***succinctly***. Show all work.

On problem	you got	out of
1		80
2		20
Σ		100

1 Short answers [80 points]

Part C.(14) You are given n activities with start times $s_1, s_2, \ldots, s_n$ and finish times $f_1 \leq f_2 \leq \ldots \leq f_n$. As we discussed you can't schedule two activities which overlap, and your goal is to schedule as many activities as possible.

However, activity r is very important, so you have to schedule it. Describe how to modify the activity selection algorithm we discussed to find the maximum size set of activities which includes activity r.

Give the running time of your algorithm.

Part B.(10) On the Homework we suggested spliting the strip into four sets of points instead of just two. Suppose now that we split the strip into 6 sets of points (each of width $d/3$). Let the $x-$coordinate of the dividing line be X. Let Y'_{L1} be the points (x, y) on the left side of the dividing line such that $d/3 \geq X - x \geq 0$ (so their $x-$coordinate is at most $d/3$ from the dividing line). Similarly, Y'_{R3} contains the points on the right such that their $x-$coordinate is more than $2d/3$ from the dividing line, but is at most d from it.

For a point (a, b) in Y'_{L1}, what is the maximum y coordinate of a point in Y'_{R3} for which we need to compute the distance from (a, b)? (drawing a picture may be helpful).

Part C.(15) Solve the following recurrences by finding a $\Theta()$ bound for $T(n)$. In each case, assume that $T(n)$ is 1 for $n \leq 1$, and otherwise is defined by the recurrence.

1. $T(n) = T(n/8) + 2T(n/7) + 5n^2$

2. $T(n) = 8T(n/2) + 5n^2$

Part D. (15) In the standard subest sum formulation we have n integers $x_1, x_2, \ldots, x_n$ and a target b. Suppose that we instead have one non-integer value x_i (e.g. $x_i = 3.4$) but the rest (including b) are all integers.

i) Describe how to modify the standard dynamic programming formulation to solve subset sum with one non-integer value.

ii) Give the run-time of your solution.

Part E. (9) Suppose we have a hash table of size m, using chaining, with n items already inserted. If we insert a new item x into this table, what is the expected number of items in the chain x hashes to (assume uniform hashing)? Briefly justify your answer.

Part F. (17) Suppose that we have an unsorted list L of $200 * n$ numbers where there are 20 copies of each item in the list (so $10n$ distinct values).

i) Suppose that we wanted to create a new list of the $10n$ distinct values in L. Describe an efficient algorithm to do this and discuss the run time.

ii) Suppose we want to find the kth element of the original list L where $k = 100 * n$ (note: this is unrelated to part i). What is the **worst case** number of calls to the partition routine used by the randomized select algorithm? Assume we use the version of the partition routine which splits the list into three groups at each partition step (those less than, equal to, and greater than the split element). Briefly justify your answer.

2 Coin Changing Varients [20 points]

Suppose we need to make change for the new country of Fubar which uses coins of value 1,5, 12 and 24 cents. As usual, we want to make chage using as few coins as possible.

Part A. Show that the greedy strategy of always using the largest coin is NOT optimal.

Part B. Describe an efficient algorithm to compute *Change(N)*, the smallest number of coins required to make change of N. Your goal is an algorithm which is as fast as possible for large N (but you needn't worry about constant factors).

Part C. Justify the correctness of your solution of part **B**.