Design & Analysis of Algorithms - Spring 2013
Final Exam

May 22, 2013 **Time: 3 hrs.** **Marks: 70**

Q1.(**5+ 5**)The following function recursively adds the elements of an array.

i. Write down the recurrence for the running time of this function on an array of size n. Also give the base case.

ii. Solve the recurrence in (i) to find a tight bound on the running time.

```
AddArrayElements(A, left, right)
 //A[1…n] is an array of integers, where n is a power of 7
size← right – left + 1
IF size > = 7
   seventh_part← FLOOR (size / 7)
   sum← 0
   FOR i←1 to 7
      sum←sum + AddArrayElements(A, left, left + seventh_part – 1)
      left← left + seventh_part
   END-FOR
ELSE
   sum← 0
   FOR i← left to right
      sum← sum + A[i]
   END-FOR
END-IF
return sum
```

Q2. (**5 + 10**)Following questions are related to the sorting algorithms covered in class.

i. A *stable* sorting algorithm leaves equal-key items in the same relative order as in the original permutation. Explain, *by providing code*, what must be done to **ensure** that mergesort is a stable sorting algorithm.

ii. Give the most efficient algorithm to find the union of sets A and B. The sizes of both A and B are the same, say n. The numbers in A and B are from the range [-k, +k] where k < n. The algorithm returns the result in a set C. Please provide a brief explanation in English, then a neat and clear pseudo code, and then an argument for the running time.

Q3.(**15**) Given a simple undirected graph G=(V, E) and an edge *e* in E, design a linear time algorithm to detect whether there is a cycle in G which contains *e* as one of its edges. The algorithm should report if no such cycle exists, otherwise it should print the cycle found (i.e. print the sequence of vertices in that cycle). Please provide a brief explanation in English, then a neat and clear pseudo code, and an argument for the running time.

Q4.(**5 * 3**) Answer the following questions, and in each case provide a brief explanation for your answer. No marks shall be awarded without a correct explanation.

i. A k-ary tree is a tree in which every node has at most k children. In a k-ary tree with n nodes and height h, what is the maximum possible number of leaves?

ii. You are given an array of size n, containing integers in the range $0 \ldots k$; and you are allowed to preprocess this array in whatever manner you choose to in any amount of time, but using only O(n) space. Then, queries of the following type are made: How many integers fall in the range $[a, b]$ (both a&b inclusive)? What is the fastest possible time in which you can answer such a query? What preprocessing will be needed in this case. Clearly explain in English.

iii. Write the equation describing the average case running time of the ***quicksort*** algorithm. State the meaning of each term in the equation and why it has been included.

iv. The shortest path tree computed by Dijkstra's algorithm is not *necessarily* an MST. Show an example graph with 4 vertices where the shortest path tree is different from the MST.

v. If we wanted to find the Maximum spanning tree of a graph, rather than the Minimum spanning tree, we could simple negate the weight on each edge and run Kruskal's algorithm, and the output would be a Maximum spanning tree. Yes or No? If yes, argue why your claim is correct. If no, give an example graph (with no more than four vertices) where this strategy will not work.

Q5. [FOR SECTIONS B, C, ONLY](5 + 5 + 5) You are going on a long trip. You start on the road at milepost $a_0 = 0$. Along the way there are *n* hotels, at mileposts $a_1 < a_2 < a_3 \ldots < a_n$, where each a_i is measured from the starting point. The only places where you are allowed to stop are these hotels, but you can choose which of these hotels to stop at. You must stop at the final hotel at a_n, which is your destination.

Ideally, you will like to travel 200 miles per day, but this may not be possible depending upon the spacing between the hotels. If you travel *x* miles a day then the penalty for that day is $(200 - x)^2$. You want to plan the trip so as to minimize the total penalty, that is, the sum of penalties over all travelling days. Your goal is to design a Dynamic Programming algorithm that gives you the optimal hotels to stop at as well as the penalty incurred with those stops.

i. The optimal structure for this problem is defined as below:
P[i] is the minimum possible penalty when the hotel at a_i is taken as destination (where you must stop).
Obviously, P[0] = 0
Give the recurrence which relates P[i] to smaller sub-problems.

ii. Let S be an array of integersof size n which stores the actual sequence of "stops" required for the optimal (minimum) penalties stored in P. Write a small piece of pseudo-code that will compute S[i] (should be on the basis of your recurrence in (i)).

iii. What is the running time of this solution in Big-O notation?

Note: In this problem you must answer part (i) correctly to get credit for parts (ii) and (iii).

Q5. [FOR SECTIONS A, D, ONLY] (15) Devise an algorithm to delete all occurrences of a pattern P (|P|=n) from Text T (|Y|=m) in O(m+n). Please provide a brief explanation in English, then a neat and clear pseudo code, and an argument for the running time.