Homework 2

Quicksort, AVL Tree, Recurrence, Dynamic Programming, Splay, DFS, Directed Acyclic Graph

1. [5 points] Analyse the expected runtime of quicksort algorithm.

2. [5 points] Using recursion-tree method solve the following recurrence

$$T(n) = 3T\left(\frac{n}{4}\right) + cn^2.$$

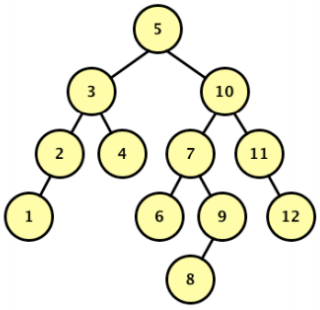
3. [4 + 1 = 5 points] Given a chain of four matrices A1, A2, A3, A4, A5, A6. The dimensions of the matrices are

   A1: 4 x 5
   A2: 5 x 8
   A3: 8 x 13
   A4: 13 x 5
   A5: 5 x 11
   A6: 11 x 7

   Solve the above matrix-chain problem using dynamic programming. Compute the minimum cost and present the parenthesization required.

4. [5 points] Show the AVL tree that results after each of the integer keys 9, 27, 50, 15, 2, 21 and 36 are inserted, in that order, into an initially empty AVL tree. Clearly show the tree results after each insertion and make clear any rotations that must be performed.

5. [5 points] Delete 3 from the given AVL tree. Show your work in detail. Note, if required use inorder successor to replace the value of the node to be deleted.



6. [2 + 3 = 5 points] What is the purpose of Splay tree? Describe delete operation in a splay tree using an example.

7. [5 points] Let S = {a, b, c, d, e, f, g} be a collection of objects with benefit-weight values as follows:

	a	b	c	d	e	f	g
Benefit	7	9	8	11	14	12	10
Weight	1	6	5	7	3	4	6

What is an optimal solution to the 0-1 Knapsack problem for S assuming we have a sack that can hold objects with total weight 18? Show your work.

8. [3 + 2 = 5 points]
   a) Write down the recurrence equation for the well-known minimum Coin Change Problem. In other words, give change for amount n using the minimum number of coins of denominations $d_1 < d_2 < \cdots < d_m$.
   Example:

   Input: coins[ ] = {5, 10, 25}, n = 30
   Output: Minimum 2 coins required
   Coin used: 5 and 25 cents

   Input: coins[ ] = {1, 5, 6, 9}, n = 11
   Output: Minimum 2 coins required
   Coin used: 5 and 6 cents

   b) Apply dynamic programming algorithm find the minimum number of coins required for the denominations 1, 3, 5 and the amount n=9. You must show your execution using a table as shown in the class.

9. [5 points] Rewrite the procedure of DFS, using a stack to eliminate recursion.

10. [5 points] Show the ordering of vertices produced by TOPOLGICAL-SORT when it is run on the following Directed Acyclic Graph (DAG), alphabetical order will be used to break ties where appropriate (e.g. so you will start from node m).