INSTRUCTIONS:

Write or type your answers on paper. Hand in your answers in class on the due date shown above. Attempt all problems. The maximum grade on this assignment is 20.

Problem 1: **[total 16 pts]** The *change making* problem is defined as follows: given is a set of coin denominations $S = \{s_1, s_2, \ldots, s_k\}$ and a sum of money V. For example, in Canada, $S = \{1, 5, 10, 25, 100, 200\}$ (pennies, nickels, dimes, quarters, loonies, and toonies). Output: a set of coins of minimum cardinality totalling V, represented by the tuple $N = (n_1, n_2, \ldots, n_k)$ where n_i represents the number of coins of denomination s_i. The total number of coins is then: $\sum_{i=1}^{k} n_i$.
For example: $v = 37$ can be represented by one 25-coin, one 10-coin and two 1-coin. The cardinality is 4 coins and is optimal. The solution is represented by tuple $(2, 0, 1, 1, 0, 0)$.

a) **[4 pts]** Describe in pseudocode a greedy algorithm for the change making problem.

b) **[2 pts]** For $S = \{1, 6, 15\}$ prove that the obvious greedy algorithm (which selects the coin with the largest denomination not larger than V) does not give the optimal solution.

c) **[5 pts]** Design a dynamic programming algorithm for the change making problem.

d) **[3 pts]** Explain how one could retrieve not only the number of coins, but also the solution (tuple N) in the dymanic programming algorithm.

e) **[2 pts]** What is the running time of the dynamic programming algorithm? Justify your answer.

Problem 2: **[4 pts]** (Exercise 4, Section 9.2) Will either Prim or Kruskal algorithms work correctly on graphs with negative edge weights? Justify your answer.