

**CS 575**  
**Homework 4**  
**Total: 100 points**

The written parts of assignment should be handed in person at the beginning of class on the submission date.

**Written Part (60 points)**

1. Not just any greedy approach to the activity-selection problem produces a maximum-size set of mutually compatible activities. Give an example to show that the approach of selecting the activity of least duration from among those that are compatible with previously selected activities does not work. Do the same for the i) approach of always selecting the compatible activity that overlaps the fewest other remaining activities and ii) always selecting the compatible remaining activity with the earliest start time. **(20 points)**

2. Consider the problem of making change for  $n$  cents using the fewest number of coins. Assume that each coin's value is an integer. **(40 points)**

- a. Describe a greedy algorithm to make change consisting of quarters, dimes, nickels, and pennies. Prove that your algorithm yields an optimal solution.
- b. Suppose that the available coins are in the denominations that are powers of  $c$ , i.e., the denominations are  $c^0, c^1, \dots, c^k$  for some integers  $c > 1$  and  $k \geq 1$ . Show that the greedy algorithm always yields an optimal solution.
- c. Give a set of coin denominations for which the greedy algorithm does not yield an optimal solution. Your set should include a penny so that there is a solution for every value of  $n$ .
- d. Give an  $O(nk)$  time algorithm that makes change for any set of  $k$  different coin denominations, assuming that one of the coins is a penny.

**Programming Part (40 points)**

**What to hand in?**

1. Submit code with in-line documentation
2. Run your code on your local machine as well as on 'remote'. A readme file outlining how your code should be run.

You are not supposed to search the Internet for code. Your solution will be compared against solutions available on the Internet and if significant match is found, you will be charged for plagiarism, in accordance to the policy outlined in the syllabus and the honesty pledge.

**You can write in the assignment in C, C++ or Java.**

**Problem 1: (20 points)**

Implement the dynamic programming algorithm described in class for solving the 0-1 Knapsack problem and show that it works correctly. You have to implement the table as described in lectures. You are required to also implement backtracking to make sure you list the set of items chosen and not just the final value.

**Problem 2: (20 points)**

a) Code the greedy algorithm you propose in Q2 a and b of the written part and show that it works for the set of inputs described in parts a and b.

b) Code the algorithm you propose in Q2 d of the written part and show that it works for any set of  $k$  different coin denominations, assuming that one of the coins is a penny.