CS3230: Assignment for Week 9

Due: Monday, 1st Nov 2021, 11:59:59 pm SGT.

Please upload PDFs containing your solutions (hand-written & scanned, or typed) by 1st Nov, 11:59:59 pm to `Assignments/Week9/Submissions`.

You may discuss the problems with your classmates, though you should write up your solutions on your own. Please note the names of your collaborators in your submission.

1. In an undirected graph G, an *independent set* is a subset S of vertices such that there is no edge between any pair of vertices in S. Consider the *maximum independent set* problem: given a graph G, find an independent set of the largest size possible.

   (a) Suppose v is a leaf node in G, meaning it has only one neighbor. Show that there is an optimal solution to the maximum independent set problem containing v.

   (b) Use the observation in part (a) to design an efficient greedy algorithm for the maximum independent set problem on trees. Explicitly describe the optimal substructure property.

2. Consider the coin-changing problem that we discussed at the end of the dynamic programming lecture. Recall that the problem is to make change for n cents using the smallest number of coins, where each coin has denomination in the set $\{d_1, \ldots, d_k\}$.

   (a) Suppose the coins are in denominations that are in powers of c. So, $d_1 = 1, d_2 = c, d_3 = c^2, \ldots, d_k = c^{k-1}$. Design and analyze a greedy algorithm to solve the problem.

   (b) Give a set of coin denominations for which the greedy algorithm from part (a) does not return the correct solution. Your set should include a 1c coin.

3. Consider the activity selection problem discussed in tutorial. Recall that in this problem, you're given a set of n activities with starting times $s_1, \ldots, s_n$ and finishing times $f_1, \ldots f_n$ where each $s_i \leq f_i$. The goal is to find the largest subset of activities which don't *conflict*, meaning that for any pair of activities selected, the finishing time of one of them is not later

than the starting time of the other. Consider each of the following greedy strategies. If it works, give a proof of the greedy-choice property. If it doesn't, show a counterexample.

(a) Choose the activity that ends first, discard all that conflict with it, and recurse.

(b) Choose an activity that conflicts with the fewest others, discard all that conflict with it, and recurse.

(c) If no activities conflict, choose them all. Otherwise, discard an activity that conflicts with the most number of other activities, and recurse.

(d) If no activities conflict, choose them all. Otherwise, discard an activity with the longest duration and recurse.