159.271 Computational Thinking for Problem Solving

Practice questions - Greedy Algorithms

1. **Coin Changing:** Suppose that one wants to make change for an amount A, using as few coins as possible, with a given set of coin denominations. A greedy algorithm for this problem could use the rule: "Select the largest coin denomination possible at each step." The rationale being that using large denominations tends to advance towards the goal A faster than using small denominations.

    1. What solution will be produced by an algorithm using this greedy rule when A = 37 and the denominations available are 1, 5, and 25?
    2. Give a counter example to show that this greedy rule does not always produce an optimal solution for denominations 1, 10 and 11.
    3. What is the worst-case running time for an algorithm applying this rule, in terms of n = | A | ?

2. **Kruskal's algorithm:** A minimal spanning tree in a connected weighted graph G is a subgraph of G that is a tree containing all of G's vertices with minimum total edge weight. Kruskal's algorithm is a greedy algorithm for finding a minimal spanning tree in a graph G. The algorithm begins with all the vertices of G and no edges. It applies the greedy rule: "Add an edge of minimum weight that does not make a cycle."

    1. Trace the solution produced by Kruskal's algorithm on the following graph. (1, 2, 3) represents an edge between vertices 1 and 2 of weight 3.
    (1, 2, 3), (1, 3, 2), (1, 4, 10), (2, 3, 9), (2, 4, 1), (3, 4, 4)
    2. Give an argument to prove the following: Suppose that G is a connected, weighted graph. If e is an edge in G whose weight is less than the weight of every other edge in G, then e is in every minimal spanning tree of G.
    Hint: Suppose that you have a spanning tree T, not containing e, then you want to show that T is not a minimum weight spanning tree.
    3. Kruskal's algorithm has the following loop invariant:
    "The current edge set (tree) contains only edges of a minimum spanning tree for each subtree constructed so far."
    Demonstrate that this invariant will hold for each iteration of the algorithm, and suggest a suitable precondition and postcondition.

3. **Prim's algorithm:** Prim's algorithm is another greedy algorithm for finding a minimal spanning tree in a graph G. The algorithm begins with a start vertex and no edges. It applies the greedy rule: "Add an edge of minimum weight that has one vertex in the current tree and the other not in the current tree."

    1. Trace the solution produced by Prim's algorithm on the following graph, with

1. Trace the solution produced by Prim's algorithm on the following graph, with start vertex 3.

   (1, 2, 3) represents an edge between vertices 1 and 2 of weight 3.

   (1, 2, 3), (1, 3, 2), (1, 4, 10), (2, 3, 9), (2, 4, 1), (3, 4, 4), (3, 5, 1)
2. With reference to the statement: "Suppose that G is a connected, weighted graph. If e is an edge in G whose weight is less than the weight of every other edge in G, then e is in every minimal spanning tree of G" - explain why any implementation of Prim's algorithm must examine each edge's weight at least once.

4. The **activity section problem** is: Given n activities and their start and finish times, find a subset of k non-conflicting activities, with k as large as possible. Two activities conflict if there is a point in time when both are active.

   Give two different greedy rules that could be used to develop a greedy algorithm to solve the activity selection problem.