CSI 503 – Data Structures and Algorithms
Pseudocode for Greedy Algorithm Examples

(a) Greedy Algorithm for Activity Scheduling:

1. Sort the tasks in increasing order of their finish times. Rename the tasks if necessary so that the sorted order of tasks is \(<a_1, a_2, \ldots, a_n>\).
2. Let \(A = \{a_1\}\). (\(A\) will contain the solution at the end of the algorithm.)
3. Let \(j = 1\). (Variable \(j\) represents the index of the last task chosen in \(A\) so that \(f_j\) will be the maximum finish time among all the tasks in \(A\).)
4. for \(i = 2\) to \(n\) do
   if \((s_i \geq f_j)\) then
      Add \(a_i\) to \(A\). (Task \(a_i\) is compatible with all the tasks in \(A\).)
      Let \(j = i\). (Now \(f_i\) is the maximum finish time among the tasks in \(A\).)
5. Output \(A\).

(b) Greedy Algorithm for Fractional Knapsack:

1. For each item \(I_j\), compute the ratio \(v_j/w_j\) (i.e., value per unit weight), \(1 \leq j \leq n\).
2. Sort the items in decreasing order of their ratios. Rename the items if necessary so that the sorted order of items is \(<I_1, I_2, \ldots, I_n>\).
3. Set TotalWeight = 0 and \(j = 1\).
4. while \((\text{TotalWeight} < W)\) do
   Let \(r = W - \text{TotalWeight}\). (Variable \(r\) represents the remaining capacity of the knapsack.)
   if \((w_j > r)\) then
      Add \(r\) lbs of item \(I_j\) to knapsack.
      Set TotalWeight = \(W\).
   else
      Add \(I_j\) to knapsack.
      Set TotalWeight = TotalWeight + \(w_j\).
      \(j = j + 1\).
5. Output the contents of Knapsack.
(c) Huffman’s Algorithm:

Given: A set $C$ of $n \geq 2$ symbols to be encoded; for each symbol $c \in C$, a frequency value $f(c)$.

Required: A codeword for each symbol in $C$ such that the cost of encoding is minimized.

Steps:

1. Create a node (of the tree) corresponding to each symbol in $C$.

2. while $(n \geq 2)$ do
   
   (a) Let $x$ be a node with smallest frequency value; remove $x$.
   (b) Let $y$ be a node with smallest frequency value; remove $y$.
   (c) Create a new node $z$ with $x$ and $y$ as its children.
   (d) Set $f(z) = f(x) + f(y)$.
   (e) Set $n = n - 1$.

3. For each internal node of the tree, label the edge to the left child as 0 and the edge to the right child as 1.

4. For each leaf $c$ (i.e., symbol in $C$), construct its codeword by concatenating the bits in the path from the root to $c$. 