Tutorial (week 11)

All the answers can be given with pseudocode or with Python. Remember that most of the time, many solutions are possible.

1. Let $S = \{a, b, c, d, e, f, g\}$ be a collection of objects with benefit-weight values, $a : (12, 4)$, $b : (10, 6)$, $c : (8, 5)$, $d : (11, 7)$, $e : (14, 3)$, $f : (7, 1)$, $g : (9, 6)$. What is an optimal solution to the fractional knapsack problem for S assuming we have a sack that can hold objects with total weight 18 ? Show your work.

2. Suppose we are given a set of tasks specified by pairs of the start times and finish times as $T = \{(1, 2), (1, 3), (1, 4), (2, 5), (3, 7), (4, 9), (5, 6), (6, 8), (7, 9)\}$. Solve the task scheduling problem for this set of tasks.

3. Draw the frequency table and Huffman tree for the following string:

   `"dogs do not spot hot pots or cats"`

4. Give an example set of 8 characters and their associated frequencies so that, in the Huffman tree for this set, every internal node has an external-node child.

5. Provide an example instance of the fractional knapsack problem where a greedy strategy based on repeatedly choosing as much of the highest-benefit item as possible results in a suboptimal solution.

6. Suppose you are given an instance of the fractional knapsack problem in which all the items have the same weight. Show that you can solve the fractional knapsack problem in this case in $O(n)$ time, instead of $O(n \log n)$.

7. A native Australian named Anatjari wishes to cross a desert carrying only a single water bottle. He has a map that marks all the watering holes along the way. Assuming he can walk k miles on one bottle of water, design an efficient algorithm for determining where Anatjari should refill his bottle in order to make as few stops as possible. Argue why your algorithm is correct.

8. Describe an efficient greedy algorithm for making change for a specified value using a minimum number of coins, assuming there are three denominations of coins (called dimes, nickels, and pennies), with values 10, 5, and 1, respectively. Argue why your algorithm is correct.

9. Give an example set of denominations of coins so that a greedy change making algorithm will not use the minimum number of coins.

10. In the art gallery guarding problem we are given a line L that represents a long hallway in an art gallery. We are also given a set $X = x_0, x_1, ..., x_{n-1}$ of real numbers that specify the positions of paintings in this hallway. Suppose that a single guard can protect all the paintings within distance at most 1 of his or her position (on both sides). Design an algorithm for finding a placement of guards that uses the minimum number of guards to guard all the paintings with positions in X.

Hints

- **Question 3: Don't forget to include the space character.**

- **Question 4: Think about powers of 2.**
- **Question 5: Think about how to create items with large weights and high benefit.**
- **Question 6: You may use the fact that it is possible to solve the selection problem for n numbers in $O(n)$ time (for example using quick-select algorithm seen in Exercise 8 of the last tutorial.)**