CSCI-SHU 220: Algorithms @ NYU Shanghai | Last Updated: November 9, 2020
Homework 4 | Due Date: November 23, 2020, 11:00pm

You are allowed to discuss with others but not allow to use references other than the course notes and reference books. Please list your collaborators for each questions. Write your own solutions and make sure you understand them.

There are 55 marks in total. The full mark of this homework is 50. Your submission should be in PDF format generated by LaTeX. You may use the LaTeX template at:

https://www.overleaf.com/read/tsxxqdgjzhxx

Enjoy :).

Problem 1: Subset sum [10 marks]

Given a set A of integers, devise an algorithm based on dynamic programming to determine whether A can be partitioned into two subsets so that the sums of the elements in both subsets are equal.

Example 1: with $A = \{1, 2, 3, 6\}$ the answer is yes,

Example 2: with $A = \{1, 2, 6\}$ the answer is no.

Problem 2: Knapsack [10 marks]

Assume that there are n objects, and the i-th object weights $w_i > 0$ kilograms and has value v_i. Here n and v_i are both positive integers with $n > v_i$. With a knapsack of capacity $W > 0$ kilograms, design an algorithm to find the maximum total value of the objects with which we can fill the knapsack. You will get full marks if the algorithm is of complexity $O(n^3)$.

Problem 3: Sorting [10 marks]

Given a sequence of numbers, $A = (a_1, \ldots, a_n)$, we will sort A in increasing order. Assume that we can insert any element of A into any position. Design an algorithm based on dynamic programming to find the minimum number of insertions that we need to sort A.

Example: with $A = (2, 3, 5, 1, 4, 7, 6)$ we need 3 insertions. First insert 1 before 2. Then insert 4 between 3 and 5. Finally insert 7 after 6 (or alternatively insert 6 before 7).

Problem 4: Covering Set [10 marks]

Given an undirected graph $G = (V, E)$, a subset $S \subseteq V$ is a covering set if for every vertex $v \in V \setminus S$, there exists $u \in S$ such that $uv \in E$ (i.e. every vertex $v \in V \setminus S$ has a neighbor in S). We are interested in finding a covering set of minimum total weight, but this problem is NP-hard in general graphs. Show that this problem is easy in trees.

- Input: A tree $T = (V, E)$ in the adjacency list representation, a weight w_v for each vertex $v \in V$.
- Output: A covering set $S \subseteq V$ that minimizes the total weight $\sum_{v \in S} w_v$.

Design an efficient algorithm to solve this problem. Prove the correctness and analyze the time complexity.

Problem 5: Selection [10 marks + 5 marks]

Let $x = (x_1, \ldots, x_m)$ and $y = (y_1, \ldots, y_k)$ be two *sorted* sequences, i.e., they satisfy

$$x_1 \leq x_2 \leq \cdots \leq x_m, \quad y_1 \leq y_2 \leq \cdots \leq y_k. \tag{1}$$

There are $\binom{m}{k}$ subsequences of x of length k, and each subsequence has $k!$ rearrangements. Denote by X the set of those rearrangements of all length-k subsequences. Note that, for each element $x' = (x_{i_1}, \ldots, x_{i_k})$ of X, the sequence x' might not be sorted, that is, it is possible that $x_{i_j} > x_{i_{j+1}}$ for some j. We can compute the squared Euclidean distance between y and x',

$$d(y, x') := \sum_{j=1}^{k} (y_j - x_{i_j})^2. \tag{2}$$

Consider the following optimization problem

$$\min_{x' \in X} \; d(y, x'). \tag{3}$$

(a) (**Bonus**: 5 marks) Prove that there some element of X which is an optimal solution to (3) and is also a sorted sequence.

(b) (10 marks) Design an algorithm based on dynamic programming to find an optimal solution to (3) (not just the optimal value).