koutecky+ads1@iuuk.mff.cuni.cz

12. tutorial

1. Counting LISs. Consider the sequence $2, 1, 3$; it has two longest increasing subsequences, namely $2, 3$ and $1, 3$. Design an algorithm which *counts* how many longest increasing subsequences are contained in an input sequence $a_1, \ldots, a_n$.

2. Maximum Independent Set. Let $T = ([n], E)$ be a tree. A subset of vertices $I \subseteq V$ is *independent* if its induced subgraph contains no edges, that is, $\forall uv \in E : u \notin I \vee v \notin I$.

(a) Design an algorithm which computes the largest independent set in T.

(b) Let $w_1, \ldots, w_n$ be vertex weights. Design an algorithm which computes an independent set I with maximum weight $\sum_{v \in I} w_v$.

3. Broadcasting in a Tree. Suppose we need to broadcast a message to all the nodes in a rooted binary tree. Initially, only the root node knows the message. In a single round, any node that knows the message can forward it to at most one of its children. Design an algorithm to compute the minimum number of rounds required to broadcast the message to all nodes.

4. Gifts. You are given a rooted tree T representing a company hierarchy, and your task is to distribute 3 types of gifts to the employees, ranked from best (gift number 1) to worst (gift number 3). No two neighbors (employee-boss pairs) can receive the same gift, and you receive a penalty of 1 for every employee who receives a better gift than their superior (because the bosses are envious).
Design an algorithm which computes an optimal (i.e., penalty minimizing) distribution of gifts.

5. Transmission. You have decyphered a secret transmission, but the spaces are missing. All is not lost – we have a dictionary containing all the words which may occur in the transmission. The task at hand is thus to split the message into the fewest possible number of dictionary words.
(For example, `ARTISTOIL` can be split as `ART·IS·TOIL`, but splitting it as `ARTIST·OIL` contains fewer words and thus is a better solution.)

6. Library 2. The problem is similar as before. However, now we are given a maximum allowed library height H, and the task is to find the minimum possible width W. If you struggle, start by assuming all books have width 1.

---

7. Average Complexity. Prove that $\Omega(\log n)$ and $\Omega(n \log n)$ comparisons are needed for searching and sorting, respectively, not only in the worst case but also on average, where the average is taken over all possible inputs. In the case of search, the average is taken over all possible x being searched; for sorting, the average is taken over all permutations.

8. Matrix search. An $n \times n$ matrix A of integers is given, and it satisfies the property that each row and column forms an increasing sequence. How to quickly find indices i, j such that $A_{i,j} = i + j$? If there are multiple solutions, it suffices to report one of them. We don't count the time needed to load the matrix into memory.
Hint: Look at the lower left corner of the matrix. What is implied by $A_{i,j} < i + j$ or $A_{i,j} > i + j$?