1. Suppose we are given a set of elephants $E = e_1, e_2, \ldots, e_n$ where $e_i = (w_i, s_i, v_i)$ represents the $i^{th}$ elephant, $w_i$ its weight, $s_i$ its intelligence and $v_i$ its value. Assume that the weights and intelligences are unique. We would like to find a subset of elephants $S \subseteq E$ for which bigger is smarter. Formally,

$$\forall i, j \in S, w_i < w_j \iff s_i < s_j$$

An equivalent way of looking at it is that if you were to sort the elephants in $S$ in increasing order by weight, then this same order would also be in increasing order with respect to intelligence. (Hint, it is useful to assume that the elephants in the solution are sorted in this way.)

The cost of the solution is the sum of the values $\sum_{i \in S} v_i$. We want to find a maximum-valued solution. Design a dynamic programming algorithm for the bigger-is-smarter elephant problem.

2. KT, Solved Exercise 1 of Chapter 6.

3. Given a bipartite graph $G = (X, Y, E)$ and two nodes $s \in X, t \in Y$, give an efficient algorithm to find all $s - t$ paths of length exactly three.