Exercise 6

Task 1
Construct an optimal binary search tree for the following elements v with probabilities $\gamma(v)$.

v	1	2	3	4	5	6	7
$\gamma(v)$	0.18	0.22	0.15	0.1	0.06	0.04	0.25

Task 2
Assume we want to construct an optimal binary search tree using the following greedy algorithm: Choose an element v for which $\gamma(v)$ is maximal as the root node and then continue recursively. Show that this approach does not always yield an optimal binary search tree.

Task 3
Assume we have a rod of length $n \in \mathbb{N}_+$ that can be cut into smaller pieces, e.g. a rod of length 3 can be cut at zero, one or two points, yielding rods of lengths $[3]$, $[1,2]$, $[2,1]$ or $[1,1,1]$. More generally, a rod cut is a sequence of positive lengths $[x_1, \ldots, x_m] \in \mathbb{N}_+^m$ with $1 \leq m \leq n$ and $\sum_{i=1}^m x_i = n$. Each rod length is assigned a price $p\colon \{1, \ldots, n\} \to \mathbb{N}$. The price of a cut is $\sum_{i=1}^m p(x_i)$. Use dynamic programming to implement an algorithm that runs in polynomial which finds a way to cut the rod such that the price is maximized.

Task 4
Let $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_n)$ be two sequences. X is a *subsequence* of Y if there are indices $1 \leq i_1 < i_2 < \cdots < i_m \leq n$ such that for all $1 \leq j \leq m$ it holds that $x_j = y_{i_j}$.
Use dynamic programming to implement an algorithm that runs in polynomial time which, given two sequences X and Y, computes the length of the longest common subsequence of X and Y.