Universität Siegen
Lehrstuhl Theoretische Informatik
Markus Lohrey

Exercise 7

Task 1
Construct an optimal binary search tree for the following elements v with probabilities $\gamma(v)$ (2 means 20% etc.).

v	1	2	3	4	5
$\gamma(v)$	2	1.5	4	1	1.5

Solution
To compute a BST with smallest weighted inner path length, we use dynamic programming. Let cost$[i, j]$ be the weighted inner path length of the optimal BST for the node set $\{i, \ldots, j\}$ with root r$[i, j]$.

$i\backslash j$	1	2	3	4	5
1	2	5	12.5	14.5	18.5
2	0	1.5	7	9	13
3		0	4	6	10
4			0	1	3.5
5				0	1.5

cost$[i, j]$

$i\backslash j$	1	2	3	4	5
1	1	1	3	3	3
2		2	3	3	3
3			3	3	3
4				4	5
5					5

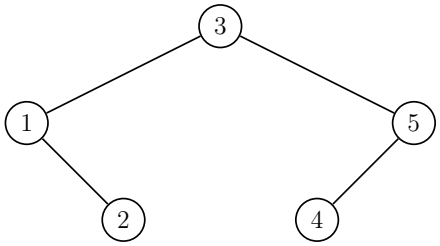
r$[i, j]$

Initialize cost$[i, i-1] = 0$, cost$[i, i] = \gamma(i)$ and r$[i, i] = i$. In the next step, the node with the highest weight (probability) has to be the root node, hence we can fill in the tables at position $(i, i+1)$. The trick in the following steps is now to pick a root, where the sum of the optimal costs of the BST for the left and the right subtree plus the total weight $\Gamma = \gamma(i) + \cdots + \gamma(j)$ is minimal (for the minimization we can ignore Γ of course). Among the optimal roots, we pick the one with the largest key (by convention).

BST of size 3: For instance $(1, 3)$; $7 + 7.5$ (1) vs. $2 + 4 + 7.5$ (2) vs. $5 + 7.5$ (3). Hence, node 3 is at the top. For the other 2 values ($(2, 4)$ and $(3, 5)$), we do the same.

BST of size 4: For instance $(2, 5)$; $10 + 8$ (2) vs. $1.5 + 3.5 + 8$ (3) vs. $7 + 1.5 + 8$ (4) vs. $9 + 8$ (5). Hence, node 3 is again at the top. The value $(1, 4)$ is obtained similarly.

BST of size 5: We have $13+10$ (1) vs. $2+10+10$ (2) vs. $5+3.5+10$ (3) vs. $12.5+1.5+10$ (4) vs. $14.5 + 10$ (5). Clearly node 3 wins. The optimal BST has weighted inner path length of 18.5 and looks like this:



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.

Solution
Choose $\gamma_1 = \frac{1}{3} - \varepsilon$, $\gamma_2 = \frac{1}{3}$ and $\gamma_3 = \frac{1}{3} + \varepsilon$. The greedy algorithm yields a chain $(3 - 2 - 1)$ with a weighted inner path length of

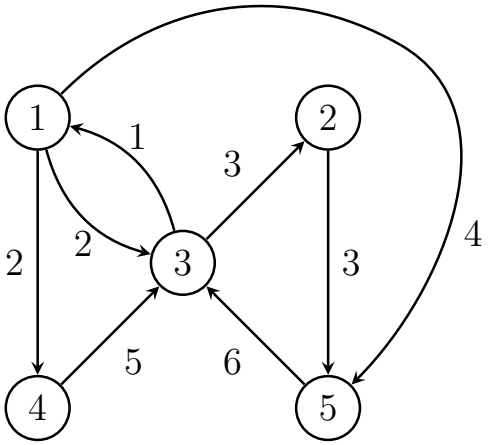
$$\left(\frac{1}{3} + \varepsilon\right) \cdot 1 + \frac{1}{3} \cdot 2 + \left(\frac{1}{3} - \varepsilon\right) \cdot 3 = 2 - 2\varepsilon.$$

It is better to take the tree with root $v = 2$ (and left child 1, right child 3). We obtain a weighted inner path length of

$$\frac{1}{3} \cdot 1 + \left(\frac{1}{3} - \varepsilon + \frac{1}{3} + \varepsilon\right) \cdot 2 = \frac{5}{3} < 2 - 2\varepsilon$$

for all $0 < \varepsilon < \frac{1}{6}$.

Task 3
Use Floyd's algorithm to compute all shortest paths in the following graph:



Solution
To use the dynamic programming approach, we compute the adjacency matrix of this graph:

$$A = \begin{pmatrix} 0 & \infty & 2 & 2 & 4 \\ \infty & 0 & \infty & \infty & 3 \\ 1 & 3 & 0 & \infty & \infty \\ \infty & \infty & 5 & 0 & \infty \\ \infty & \infty & 6 & \infty & 0 \end{pmatrix}$$

In the first step ($k = 1$) we consider the first row and first column of the matrix and change value $A[3, 4]$ and $A[3, 5]$.

$$\begin{pmatrix} 0 & \infty & 2 & 2 & 4 \\ \infty & 0 & \infty & \infty & 3 \\ 1 & 3 & 0 & 3 & 5 \\ \infty & \infty & 5 & 0 & \infty \\ \infty & \infty & 6 & \infty & 0 \end{pmatrix}$$

We proceed with $k = 2$, but nothing changes. But $k = 3$ yields:

$$\begin{pmatrix} 0 & 5 & 2 & 2 & 4 \\ \infty & 0 & \infty & \infty & 3 \\ 1 & 3 & 0 & 3 & 5 \\ 6 & 8 & 5 & 0 & 10 \\ 7 & 9 & 6 & 9 & 0 \end{pmatrix}$$

For $k = 4$ nothing changes again. Now $k = 5$ (last row, last column):

$$\begin{pmatrix} 0 & 5 & 2 & 2 & 4 \\ 10 & 0 & 9 & 12 & 3 \\ 1 & 3 & 0 & 3 & 5 \\ 6 & 8 & 5 & 0 & 10 \\ 7 & 9 & 6 & 9 & 0 \end{pmatrix}$$

This is the final matrix. Every entry $A[i, j]$ shows us exactly the shortest paths from i to j. For instance from 2 to 1 we pass by node 5 and 3 (length $3 + 6 + 1 = 10$).