These are meant as guided exercises to reinforce the algorithms from lecture. You are encouraged to understand how the algorithms from class work. Rather than memorizing the code, understand how each works and which decisions are made at each stage.

These are deliberately more time consuming than the test questions I plan to ask on this topic.

1. Consider the following table, which is the output from running the optimal binary search tree algorithm for some 11-key input. In each entry, the value printed in the upper half of the cell is OPT$[i, j]$, the cost of the optimal binary search tree consisting of keys $i \ldots j$. The value printed in the lower half of the cell is the value of roots$[i, j]$.

   Suppose we have computed the following using the algorithm for the optimal binary search tree. However, we accidentally forgot to record OPT$[1, n]$. What is the missing value for that spot? It is the only one omitted in the following table.

	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	k_{10}	k_{11}
k_1	0.11 1	0.36 2	0.42 2	0.63 2	0.84 2	1.19 2	1.59 4	1.99 5	2.16 6	2.42 6	 6
k_2		0.14 2	0.2 2	0.41 2	0.6 4	0.87 4	1.26 4	1.65 6	1.81 6	2.07 6	2.4 7
k_3			0.03 3	0.15 4	0.29 4	0.54 5	0.84 6	1.2 6	1.35 7	1.61 7	1.93 7
k_4				0.09 4	0.23 4	0.45 5	0.75 6	1.08 7	1.23 7	1.49 7	1.81 7
k_5					0.07 5	0.24 6	0.5 6	0.78 7	0.93 7	1.19 7	1.48 8
k_6						0.1 6	0.33 7	0.57 7	0.72 7	0.97 8	1.24 8
k_7							0.13 7	0.37 7	0.48 8	0.67 8	0.94 8
k_8								0.12 8	0.22 8	0.41 8	0.64 10
k_9									0.05 9	0.17 10	0.35 10
k_{10}										0.07 10	0.23 11
k_{11}											0.09 11

   2.78

2. Consider the following output for the Optimal BST algorithm, but with a different input from the previous problem. Unfortunately this time, I forgot to write down the root values! What is the root of the overall tree? That is, what should be in roots$[1, n]$?

	k_1	k_2	k_3	k_4	k_5	k_6
k_1	0.19	0.59	0.95	1.37	1.92	2.26
k_2		0.21	0.57	0.88	1.36	1.69
k_3			0.18	0.46	0.84	1.17
k_4				0.14	0.45	0.67
k_5					0.17	0.39
k_6						0.11

3