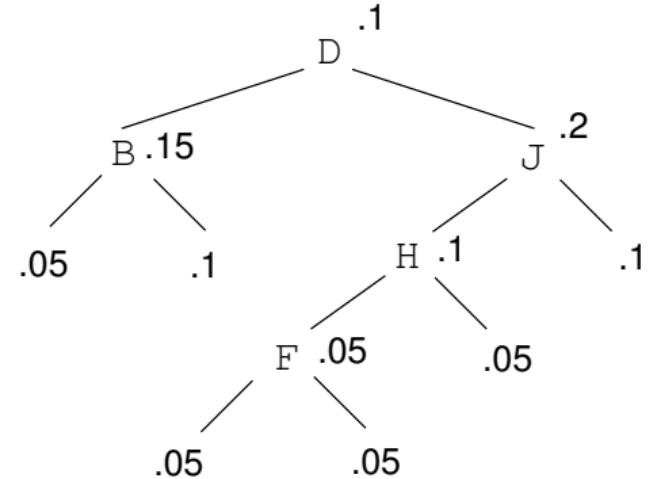
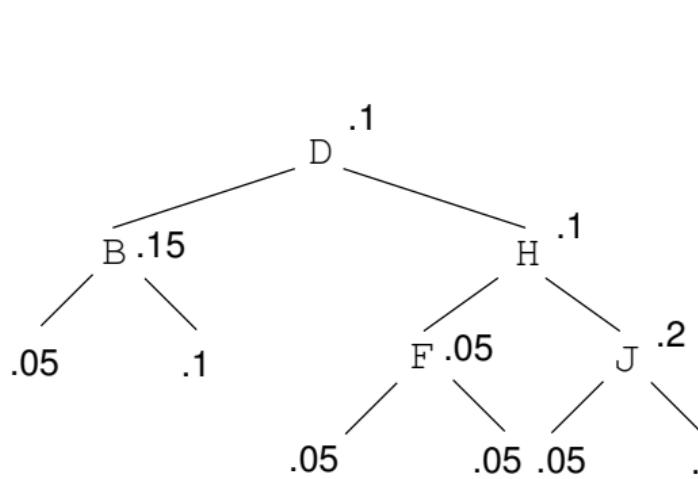


	0	1	2	3	4
Keys (k_i)	B	D	F	H	J
Key probabilities (p_i)	.15	.1	.05	.1	.2
Miss probabilities (q_i)	.05	.1	.05	.05	.05
	0	1	2	3	4

Taken from Cormen et al, *Introduction to Algorithms*, pg 397 et seq.

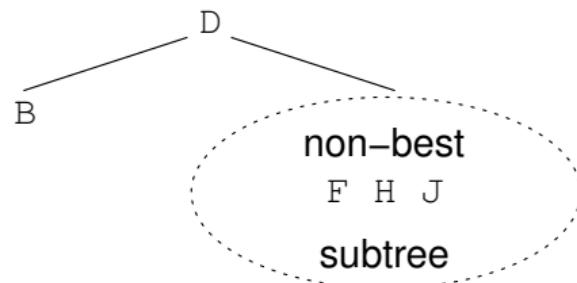
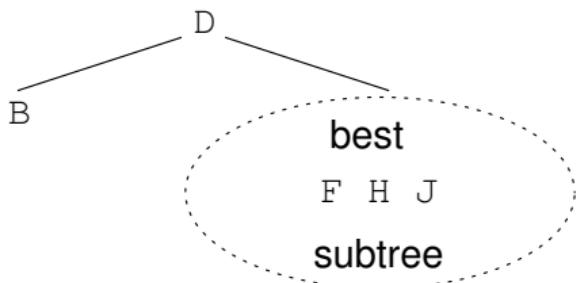
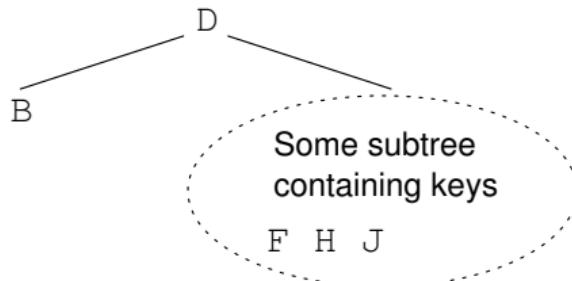


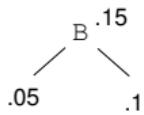
For the left tree:

$$.1 + 2(.15 + .1) + 3(.05 + .1 + .05 + .2) + 4(.05 + .05 + .05 + .1) = 2.8$$

For the right tree:

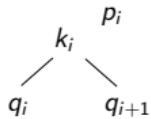
$$.1 + 2(.15 + .2) + 3(.05 + .1 + .1 + .1) + 4(.05 + .05) + 5(.05 + .05) = 2.75$$





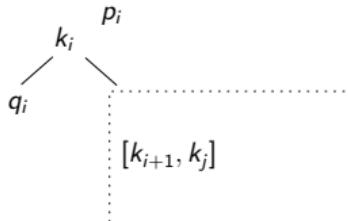
$$\text{total probability: } .05 + .15 + .1 = .3$$

$$\text{total weighted depth: } 2 \cdot .05 + .15 + 2 \cdot .1 = .45$$



$$\text{total probability: } q_i + p_i + q_{i+1}$$

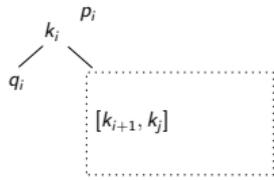
$$\text{total weighted depth: } 2 \cdot q_i + p_i + 2 \cdot q_{i+1}$$



$$\text{total probability: } q_i + p_i + \text{total prob } [k_{i+1}, k_j]$$

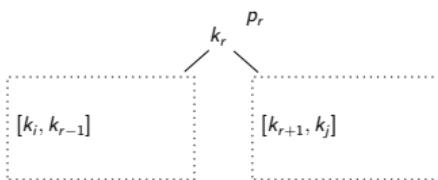
$$\text{total weighted depth: } 2 \cdot q_i + p_i + ???$$

$$\begin{aligned}
& \sum_{x=i}^j p_x \cdot \text{depth}_1(k_x) + \sum_{x=i}^{j+1} q_x \cdot \text{depth}_1(d_x) \\
\\
&= 2 \cdot q_i + p_i + \sum_{x=i+1}^j p_x \cdot \text{depth}_1(k_x) + \sum_{x=i+1}^{j+1} q_x \cdot \text{depth}_1(d_x) \\
\\
&= 2 \cdot q_i + p_i + \sum_{x=i+1}^j p_x \cdot (\text{depth}_0(k_x) + 1) + \sum_{x=i+1}^{j+1} q_x \cdot (\text{depth}_0(d_x) + 1) \\
\\
&= 2 \cdot q_i + p_i + \sum_{x=i+1}^j p_x + \sum_{x=i+1}^{j+1} q_x + \sum_{x=i+1}^j p_x \cdot \text{depth}_0(k_x) + \sum_{x=i+1}^{j+1} q_x \cdot \text{depth}_0(d_x) \\
\\
&= q_i + q_i + p_i + \sum_{x=i+1}^j p_x + \sum_{x=i+1}^{j+1} q_x + (\text{total weighted depth of subtree}) \\
\\
&= q_i + (\text{total prob of tree}) + (\text{total weighted depth of subtree})
\end{aligned}$$



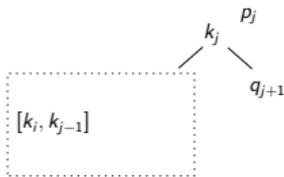
total probability: $q_i + p_i + T[i + 1][j]$

total weighted depth: $q_i + T[i][j] + C[i + 1][j]$



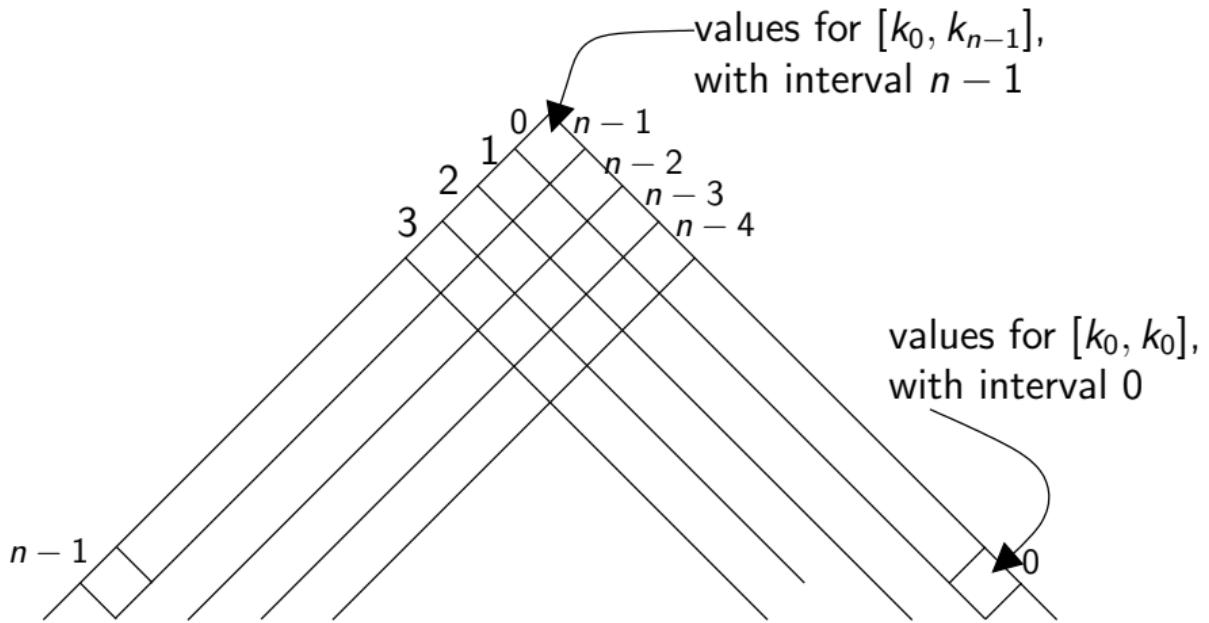
total probability: $T[i][r - 1] + p_r + T[r + 1][j]$

total weighted depth: $C[i][r - 1] + T[i][j] + C[r + 1][j]$



total probability: $T[i][j - 1] + p_j + q_{j+1}$

total weighted depth: $C[i][j - 1] + T[i][j] + q_{j+1}$



For each diagonal from bottom up

For each (i, j) in that diagonal

Determine $T[i][j]$, $C[i][j]$, and the root of the best tree

Initialize the cells $(0, 0)$ through $(n - 1, n - 1)$

For each interval size from 1 to $n - 1$

For each (i, j) in that interval ($j = i + \text{interval}$)

Find $T[i][j]$

Consider each key k_r ,

keeping track of the root of the best tree seen so far
and associated total weighted depth

Special case for k_i ($r = i$)

Compute total weighted depth, assume it's best so far

For each $k_r \in [k_{i+1}, \dots, k_{j-1}]$ (each $r \in [i + 1, \dots, j - 1]$)

Compute total weighted depth, compare with best so far

Special case for k_j ($r = j$)

Compute total weighted depth, compare with best so far

Enter table entries for (i, j)

Return tree rooted at cell $(0, n - 1)$ in node tree