Chapter 5, Dynamic Programming:

- Introduction and sample problems (previous week Wednesday)
- Principles of DP (previous week Friday)
- DP algorithms, solutions to sample problems (last week Monday)
- Introduce optimal BSTs / review for test 2 (last week Wednesday)
- **Test 2**, *not* covering DP (last week Friday)
- Retrospective on Test 2 (Monday)
- Finish up optimal BSTs (**Today**)
- [Begin hash tables (Friday)]

Today:

- Review optimal BST definition
- The optimal-BST-building problem
- The dynamic programming solution
Why this problem?

- It connects dynamic programming with the quest for a better map.
- Its hardness is in the right places (building the table—hard; reconstructing solution—trivial).
- It is a representative of a bigger concept: What if we had more information—how would that change the problem.

Game plan:

- Understand the problem itself
- Understand the recursive characterization
- Understand the table-building algorithm
The **optimal binary search tree** problem:

- Assume we know all the keys $k_0, k_1, \ldots k_{n-1}$ ahead of time.
- Assume further that we know the probabilities $p_0, p_1, \ldots p_{n-1}$ of each key’s lookup.
- Assume even further that we know the “miss probabilities” $q_0, q_1, \ldots q_n$ where $q_i$ is the probability that an extraneous key falling between $k_{i-1}$ and $k_i$ will be looked up.
- We want to build a tree to minimize the *expected cost* of a look up, which is the *total weighted depth* of the tree:

$$
\sum_{i=0}^{n-1} p_i \text{depth}(k_i) + \sum_{i=0}^{n} q_i \text{depth}(m_i)
$$

where $\text{depth}(x)$ is the number of nodes to be inspected on the route from the root to node $x$, $k_i$ stands for the node containing key $k_i$ [notational abuse], and $m_i$ is the dummy node between keys $k_{i-1}$ and $k_i$.
- Note that the rules of probability require $\sum_{i=0}^{n-1} p_i + \sum_{i=0}^{n} q_i = 1$
<table>
<thead>
<tr>
<th>Word</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>84</td>
</tr>
<tr>
<td>not</td>
<td>83</td>
</tr>
<tr>
<td>them</td>
<td>61</td>
</tr>
<tr>
<td>a</td>
<td>59</td>
</tr>
<tr>
<td>like</td>
<td>44</td>
</tr>
<tr>
<td>in</td>
<td>40</td>
</tr>
<tr>
<td>do</td>
<td>36</td>
</tr>
<tr>
<td>you</td>
<td>34</td>
</tr>
<tr>
<td>would</td>
<td>26</td>
</tr>
<tr>
<td>and</td>
<td>24</td>
</tr>
<tr>
<td>eat</td>
<td>24</td>
</tr>
<tr>
<td>will</td>
<td>21</td>
</tr>
<tr>
<td>sam</td>
<td>19</td>
</tr>
<tr>
<td>with</td>
<td>19</td>
</tr>
<tr>
<td>am</td>
<td>16</td>
</tr>
<tr>
<td>could</td>
<td>14</td>
</tr>
<tr>
<td>here</td>
<td>11</td>
</tr>
<tr>
<td>the</td>
<td>11</td>
</tr>
<tr>
<td>eggs</td>
<td>10</td>
</tr>
<tr>
<td>box</td>
<td>7</td>
</tr>
<tr>
<td>car</td>
<td>7</td>
</tr>
<tr>
<td>ham</td>
<td>10</td>
</tr>
<tr>
<td>there</td>
<td>9</td>
</tr>
<tr>
<td>train</td>
<td>9</td>
</tr>
<tr>
<td>anywhere</td>
<td>8</td>
</tr>
<tr>
<td>house</td>
<td>8</td>
</tr>
<tr>
<td>mouse</td>
<td>8</td>
</tr>
<tr>
<td>or</td>
<td>8</td>
</tr>
<tr>
<td>box</td>
<td>7</td>
</tr>
<tr>
<td>car</td>
<td>7</td>
</tr>
<tr>
<td>fox</td>
<td>7</td>
</tr>
<tr>
<td>on</td>
<td>7</td>
</tr>
<tr>
<td>tree</td>
<td>6</td>
</tr>
<tr>
<td>see</td>
<td>4</td>
</tr>
<tr>
<td>try</td>
<td>4</td>
</tr>
<tr>
<td>boat</td>
<td>3</td>
</tr>
<tr>
<td>say</td>
<td>5</td>
</tr>
<tr>
<td>so</td>
<td>5</td>
</tr>
<tr>
<td>that</td>
<td>3</td>
</tr>
<tr>
<td>be</td>
<td>4</td>
</tr>
<tr>
<td>are</td>
<td>2</td>
</tr>
<tr>
<td>good</td>
<td>2</td>
</tr>
<tr>
<td>goat</td>
<td>4</td>
</tr>
<tr>
<td>may</td>
<td>4</td>
</tr>
<tr>
<td>they</td>
<td>2</td>
</tr>
<tr>
<td>me</td>
<td>4</td>
</tr>
<tr>
<td>if</td>
<td>1</td>
</tr>
<tr>
<td>Key or miss event</td>
<td>combined frequency</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>{}</td>
<td>0</td>
</tr>
<tr>
<td>a</td>
<td>59</td>
</tr>
<tr>
<td>{ am and anywhere are be boat box car could dark }</td>
<td>92</td>
</tr>
<tr>
<td>do</td>
<td>36</td>
</tr>
<tr>
<td>{ eat eggs fox goat good green ham here house }</td>
<td>86</td>
</tr>
<tr>
<td>i</td>
<td>84</td>
</tr>
<tr>
<td>{ if let }</td>
<td>5</td>
</tr>
<tr>
<td>in</td>
<td>40</td>
</tr>
<tr>
<td>{}</td>
<td>0</td>
</tr>
<tr>
<td>like</td>
<td>44</td>
</tr>
<tr>
<td>{ may me mouse }</td>
<td>16</td>
</tr>
<tr>
<td>not</td>
<td>83</td>
</tr>
<tr>
<td>{ on or rain same say see so thank that the }</td>
<td>65</td>
</tr>
<tr>
<td>then</td>
<td>61</td>
</tr>
<tr>
<td>{ there they train tree try will with would }</td>
<td>99</td>
</tr>
<tr>
<td>you</td>
<td>34</td>
</tr>
<tr>
<td>{}</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>---</td>
<td>-----</td>
</tr>
<tr>
<td>$k_i$</td>
<td>do</td>
</tr>
<tr>
<td>$p_i$</td>
<td>.073</td>
</tr>
<tr>
<td>$q_i$</td>
<td>.001</td>
</tr>
</tbody>
</table>
1 \cdot .02 + 1 \cdot .081 = .101
\[2 \cdot .02 + 2 \cdot .081 + 1 \cdot .103 + 1 \cdot .122 = .427\]
\(3 \cdot .02 + 3 \cdot .081 + 2 \cdot .103 + 2 \cdot .122 + 1 \cdot .001 + 1 \cdot .133 + 1 \cdot .107 + 1 \cdot .006 + 1 \cdot .076 + 1 \cdot .001 = 1.057\)
\[ 4 \cdot .02 + 4 \cdot .081 + 3 \cdot .103 + 3 \cdot .122 + 2 \cdot .001 + 2 \cdot .133 + 2 \cdot .107 + 2 \cdot .006 + 2 \cdot .076 + 2 \cdot .001 + 1 \cdot .073 + 1 \cdot .104 + 1 \cdot .001 + 1 \cdot .042 = 1.907 \]
5 \cdot 0.02 + 5 \cdot 0.081 \\
+ 4 \cdot 0.103 + 4 \cdot 0.122 \\
+ 3 \cdot 0.001 + 3 \cdot 0.133 + 3 \cdot 0.107 + 3 \cdot 0.006 + 3 \cdot 0.076 + 3 \cdot 0.001 \\
+ 2 \cdot 0.073 + 2 \cdot 0.104 + 2 \cdot 0.001 + 2 \cdot 0.042 \\
+ 1 \cdot 0.045 + 1 \cdot 0.055 \\
= 2.857
\[
6 \cdot 0.02 + 6 \cdot 0.081 \\
+ 5 \cdot 0.103 + 5 \cdot 0.122 \\
+ 4 \cdot 0.001 + 4 \cdot 0.133 + 4 \cdot 0.107 + 4 \cdot 0.006 + 4 \cdot 0.076 + 4 \cdot 0.001 \\
+ 3 \cdot 0.073 + 3 \cdot 0.104 + 3 \cdot 0.001 + 3 \cdot 0.042 \\
+ 2 \cdot 0.045 + 2 \cdot 0.055 \\
+ 1 \cdot 0.05 \\
= 3.857
\]
$$4 \cdot .001 + 3 \cdot .073 + 4 \cdot .133 + 2 \cdot .045 + 4 \cdot .107 + 3 \cdot .104 + 4 \cdot .006 \ + 1 \cdot .05 \ + 3 \cdot .001 + 2 \cdot .055 + 6 \cdot .02 + 6 \cdot .081 + 4 \cdot .076 + 5 \cdot .122 + 3 \cdot .042 + 4 \cdot .001$$
$$= 3.857$$
\[ 3 \cdot .001 + 2 \cdot .073 + 3 \cdot .133 + 1 \cdot .045 + 3 \cdot .107 + 2 \cdot .104 + 3 \cdot .006 \\
+ 2 \cdot .001 + 1 \cdot .055 + 5 \cdot .02 + 5 \cdot .081 + 3 \cdot .076 + 4 \cdot .122 + 2 \cdot .042 + 3 \cdot .001 \\
+ .001 + .073 + .133 + .045 + .107 + .104 + .006 \\
+ .05 \\
+ .001 + .055 + .02 + .081 + .076 + .122 + .042 + .001 \\
= 3.857 \\
\]

\[
\text{in .05}
\]

\[
\begin{array}{c}
1 \\
\text{do .045} \\
2 \\
a .073 \\
3 \\
\cdot .001 \\
3 \\
1 \\
\text{like .055} \\
2 \\
i .104 \\
3 \\
\cdot .113 \\
3 \\
3 \\
\cdot .107 \\
3 \\
3 \\
\cdot .006 \\
3 \\
\cdot .001 \\
2 \\
you .042 \\
3 \\
\cdot .076 \\
3 \\
3 \\
\cdot .001 \\
3 \\
\cdot .103 \\
4 \\
\cdot .122 \\
4 \\
5 \\
\cdot .02 \\
5 \\
\cdot .081 \\
4 \\
\end{array}
\]
Total weighted depth for a given tree (expected lookup cost):

\[
\sum_{i=0}^{n-1} p_i \text{depth}(k_i) + \sum_{i=0}^{n} q_i \text{depth}(m_i)
\]

Let \( \text{depth}_{k_a}(k_i) \) be the depth of the node with \( k_i \) in the subtree rooted at node with \( k_1 \). For example, if \( k_r \) is the root of the entire tree and \( k_a \) is a child of the root, then

\[
\text{depth}_{k_r}(k_i) = \text{depth}_{k_a}(k_i) + 1
\]

Then we can rewrite the total weighted depth as

\[
\sum_{i=0}^{r-1} p_i \text{depth}_{k_r}(k_i) + \sum_{i=0}^{r} q_i \text{depth}_{k_r}(m_i) + p_r + \sum_{i=r+1}^{n-1} p_i \text{depth}_{k_r}(k_i) + \sum_{i=r+1}^{n} q_i \text{depth}_{k_r}(m_i)
\]

left subtree total weighted depth (absolute)  right subtree total weighted depth (absolute)
Again, let $k_r$ be the root of the entire tree and $k_a$ and $k_b$ be the root's children. Then

\[
\sum_{i=0}^{r-1} p_i (\text{depth}_{k_a}(k_i) + 1) + \sum_{i=0}^r q_i (\text{depth}_{k_a}(m_i) + 1) + p_r + \sum_{i=r+1}^{n-1} p_i (\text{depth}_{k_b}(k_i) + 1) + \sum_{i=r+1}^n q_i (\text{depth}_{k_r}(m_i) + 1)
\]

\[\text{left subtree total weighted depth (absolute)} + \text{right subtree total weighted depth (absolute)}\]

Convert to “relative depth”:

\[
\sum_{i=0}^{n-1} p_i + \sum_{i=0}^n q_i + \sum_{i=0}^{r-1} p_i \text{ depth}_{k_a}(k_i) + \sum_{i=0}^r q_i \text{ depth}_{k_a}(m_i) + \sum_{i=r+1}^{n-1} p_i \text{ depth}_{k_b}(k_i) + \sum_{i=r+1}^n q_i \text{ depth}_{k_r}(m_i)
\]

\[\text{total probability} + \text{left subtree total weighted depth (relative)} + \text{right subtree total weighted depth (relative)}\]

Let $TWD(k)$ be the total weighted depth of the tree rooted at $k$ (relative to $k$) and $TP(k)$ be the total probability of the tree rooted at $k$. Then

\[
TWD(k_r) = TP(k_r) + TWD(k_a) + TWD(k_b)
\]
Let $P[i][j]$ be the total probabilities of the keys and misses in the range $[i, j]$:

$$P[i][j] = \sum_{k=i}^{j} p_k + \sum_{k=i}^{j+1} q_k$$

Let $C[i][j]$ be the least total weighted depth of any BST composed from keys in the range $[i, j]$. The recursive characterization is

$$C[i][j] = \begin{cases} 
2q_i + p_i + 2q_{i+1} & \text{if } i = j \\
2q_i + p_i + 2q_{i+1} + \min \left\{ q_i + C[i+1][j] , \right. \\
& \left. C[i][r-1] + C[r+1][j] \text{ for } r \in (i, j) \right\} & \text{if } i < j
\end{cases}$$
\[ C[i][j] = \begin{cases} 2q_i + p_i + 2q_{i+1} & \text{if } i = j \\ P[i][j] + \min \left\{ \begin{array}{ll} q_i + C[i + 1][j] \\ C[i][r - 1] + C[r + 1][j] \text{ for } r \in (i, j) \\ C[i][j - 1] + q_{j+1} \end{array} \right. & \text{if } i < j \end{cases} \]
\[ C[i][j] = \begin{cases} 
2q_i + p_i + 2q_{i+1} & \text{if } i = j \\
P[i][j] + \min \left\{ \begin{array}{l}
q_i + C[i+1][j] \\
C[i][r-1] + C[r+1][j] & \text{for } r \in (i, j) \\
C[i][j-1] + q_{j+1}
\end{array} \right\} & \text{if } i < j
\end{cases} \]
\[
C[i][j] = \begin{cases} 
2q_i + p_i + 2q_{i+1} & \text{if } i = j \\
P[i][j] + \min \left\{ \begin{array}{l} 
q_i + C[i+1][j] \\
C[i][r-1] + C[r+1][j] \text{ for } r \in (i, j) \\
C[i][j-1] + q_{j+1} 
\end{array} \right\} & \text{if } i < j 
\end{cases}
\]

\[
P[i][j] = \begin{cases} 
q_i + p_i + q_{i+1} & \text{if } i = j \\
\left\{ \begin{array}{l} 
q_i + p_i + P[i+1][j] \\
or \quad P[i][r-1] + p_r + P[r+1][j] \text{ for } r \in (i, j) \\
or \quad P[i][j-1] + p_j + q_{j+1} 
\end{array} \right\} & \text{if } i < j 
\end{cases}
\]
\[
\begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
k_i & a & d & o & i & i & n & k & l & e & n & t & h & e & n & t & y \\
p_i & .073 & .045 & .104 & .05 & .055 & .103 & .076 & .042 \\
q_i & .001 & .113 & .107 & .006 & .001 & .02 & .081 & .122 & .001 \\
\end{array}
\]
Coming up:

Due Wed, Nov 16 (end of day)
Read Section 6.5
(No quiz on Section 6.5)

Due Fri, Nov 18 (end of day)
Read Sections 7.(1 & 2)
Take quiz

Due Mon, Nov 21 (end of day)
Do Project 7.1 (as practice problem)

Due Mon, Nov 28 (end of day) (recommended to be done before break)
Read Section 7.3
Do Exercises 7.(4,5,7,8)
Take quiz

Do Optimal BST project (suggested by Monday, Nov 21)