Chapter 5, Dynamic Programming:

- ▶ Introduction and sample problems (last week Friday)
- ▶ Principles of DP (Monday)
- ▶ DP algorithms, solutions to sample problems (Wednesday)
- ▶ Begin optimal BSTs (**Today**)
- ▶ Finish optimal BSTs/review for test (next week Monday)
- ▶ [Test 2, Wed, Apr 5, *not* covering DP]

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:

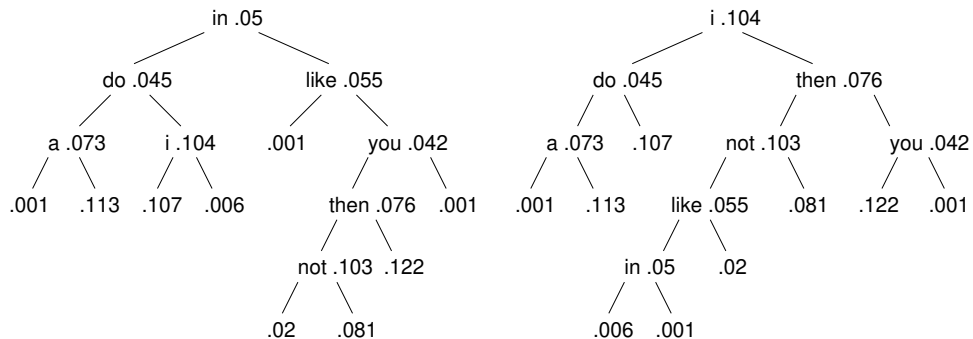$$\sum_{i=0}^{n-1} p_i \; depth(k_i) + \sum_{i=0}^{n} q_i \; depth(m_i)$$

  where $depth(x)$ 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 and k_i.
- ▶ Note that the rules of probability require $\sum_{i=0}^{n-1} p_i + \sum_{i=0}^{n} q_i = 1$

i	84	eat	24	ham	10	fox	7	rain	4
not	83	will	21	there	9	on	7	see	4
them	61	sam	19	train	9	tree	6	try	4
a	59	with	19	anywhere	8	say	5	boat	3
like	44	am	16	house	8	so	5	that	3
in	40	could	14	mouse	8	be	4	are	2
do	36	here	11	or	8	goat	4	good	2
you	34	the	11	box	7	let	4	thank	2
would	26	eggs	10	car	7	may	4	they	2
and	24	green	10	dark	7	me	4	if	1

Key or miss event	combined frequency
{ }	0
a	59
{ am and anywhere are be boat box car could dark }	92
do	36
{ eat eggs fox goat good green ham here house }	86
i	84
{ if let }	5
in	40
{ }	0
like	44
{ may me mouse }	16
not	83
{ on or rain same say see so thank that the }	65
then	61
{ there they train tree try will with would }	99
you	34
{ }	0

	0	1	2	3	4	5	6	7
k_i	a	do	i	in	like	not	then	you
p_i	.073	.045	.104	.05	.055	.103	.076	.042
q_i	.001	.113	.107	.006	.001	.02	.081	.122	.001

in .05
  do .045
    a .073
      .001   .113
    i .104
      .107   .006
  like .055
    .001
    you .042
      then .076
        not .103
          .02   .081
        .122
      .001

i .104
  do .045
    a .073
      .001   .113
    .107
  then .076
    not .103
      like .055
        in .05
          .006   .001
        .02
      .081
    you .042
      .122   .001

$1 \cdot .02 + 1 \cdot .081$
$= .101$

$$1 \qquad\qquad 1$$
$$.02 \qquad .081$$

$2 \cdot .02 + 2 \cdot .081$
$+1 \cdot .103 + 1 \cdot .122$
$= .427$
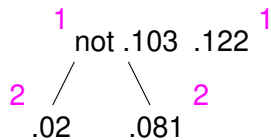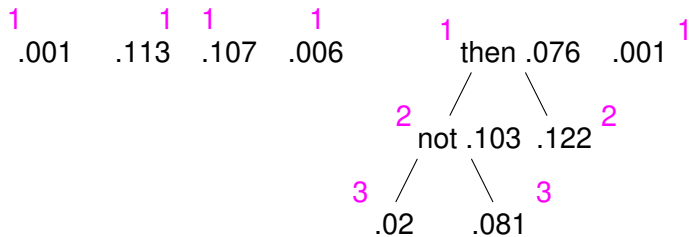
$3 \cdot .02 + 3 \cdot .081$
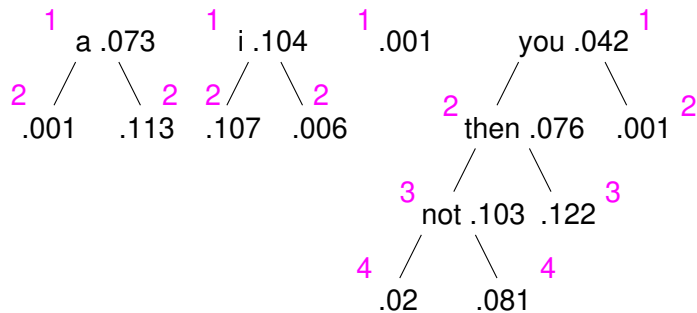$+2 \cdot .103 + 2 \cdot .122$
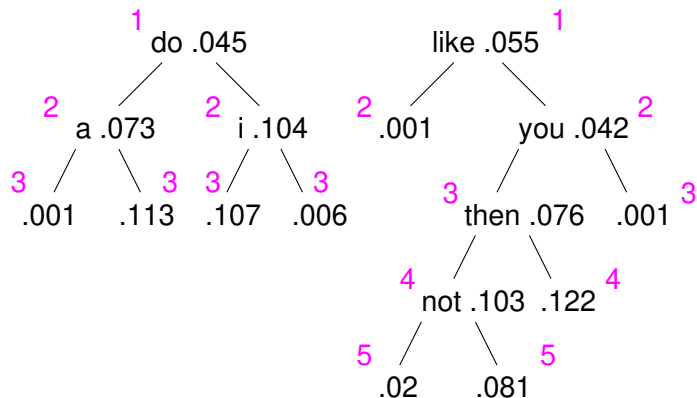$+1 \cdot .001 + 1 \cdot .133 + 1 \cdot .107 + 1 \cdot .006 + 1 \cdot .076 + 1 \cdot .001$
$= 1.057$

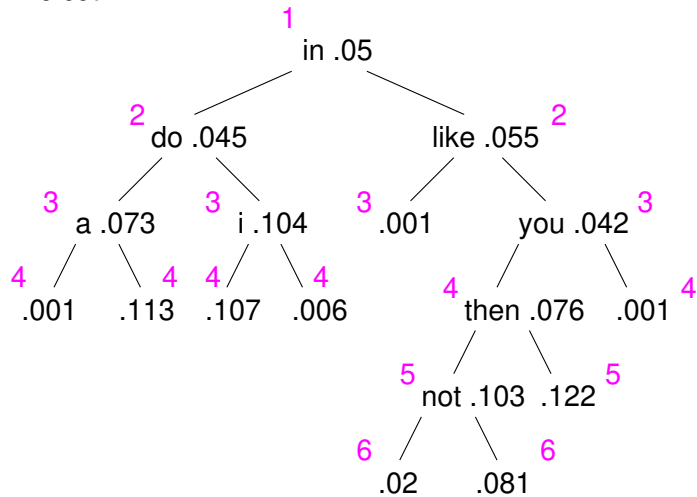$$4 \cdot .02 + 4 \cdot .081$$
$$+3 \cdot .103 + 3 \cdot .122$$
$$+2 \cdot .001 + 2 \cdot .133 + 2 \cdot .107 + 2 \cdot .006 + 2 \cdot .076 + 2 \cdot .001$$
$$+1 \cdot .073 + 1 \cdot .104 + 1 \cdot .001 + 1 \cdot .042$$
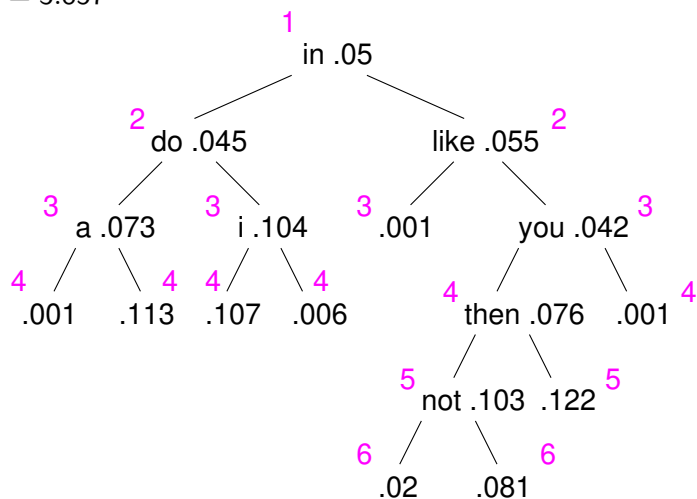$$= 1.907$$

$5 \cdot .02 + 5 \cdot .081$
$+ 4 \cdot .103 + 4 \cdot .122$
$+ 3 \cdot .001 + 3 \cdot .133 + 3 \cdot .107 + 3 \cdot .006 + 3 \cdot .076 + 3 \cdot .001$
$+ 2 \cdot .073 + 2 \cdot .104 + 2 \cdot .001 + 2 \cdot .042$
$+ 1 \cdot .045 + 1 \cdot .055$
$= 2.857$

$6 \cdot .02 + 6 \cdot .081$

$+5 \cdot .103 + 5 \cdot .122$

$+4 \cdot .001 + 4 \cdot .133 + 4 \cdot .107 + 4 \cdot .006 + 4 \cdot .076 + 4 \cdot .001$

$+3 \cdot .073 + 3 \cdot .104 + 3 \cdot .001 + 3 \cdot .042$
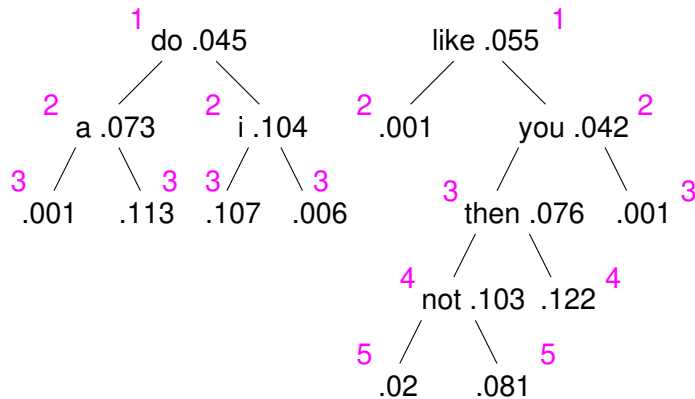
$+2 \cdot .045 + 2 \cdot .055$

$+1 \cdot .05$

$= 3.857$

$4 \cdot .001 + 3 \cdot .073 + 4 \cdot .133 + 2 \cdot .045 + 4 \cdot .107 + 3 \cdot .104 + 4 \cdot .006$

$+ 1 \cdot .05$

$+ 3 \cdot .001 + 2 \cdot .055 + 6 \cdot .02 + 6 \cdot .081 + 4 \cdot .076 + 5 \cdot .122 + 3 \cdot .042 + 4 \cdot .001$

$= 3.857$

$3 \cdot .001 + 2 \cdot .073 + 3 \cdot .133 + 1 \cdot .045 + 3 \cdot .107 + 2 \cdot .104 + 3 \cdot .006$
$+2 \cdot .001 + 1 \cdot .055 + 5 \cdot .02 + 5 \cdot .081 + 3 \cdot .076 + 4 \cdot .122 + 2 \cdot .042 + 3 \cdot .001$
$+.001 + .073 + .133 + .045 + .107 + .104 + .006$
$+.05$
$+.001 + .055 + .02 + .081 + .076 + .122 + .042 + .001$
$= 3.857$

in .05

Total weighted depth for a given tree (expected lookup cost):

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

Let $depth_{k_a}(k_i)$ 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

$$depth_{k_r}(k_i) = depth_{k_a}(k_i) + 1$$

Then we can rewrite the total weighted depth as

$$\underbrace{\sum_{i=0}^{r-1} p_i \ depth_{k_r}(k_i) + \sum_{i=0}^{r} q_i \ depth_{k_r}(m_i)}_{\text{left subtree total weighted depth (absolute)}} + p_r + \underbrace{\sum_{i=r+1}^{n-1} p_i \ depth_{k_r}(k_i) + \sum_{i=r+1}^{n} q_i \ depth_{k_r}(m_i)}_{\text{right subtree total weighted depth (absolute)}}$$

Again, let k_r be the root of the entire tree and k_a and k_b be the root's children. Then

$$\underbrace{\sum_{i=0}^{r-1} p_i(depth_{k_a}(k_i)+1) + \sum_{i=0}^{r} q_i(depth_{k_a}(m_i)+1)}_{\text{left subtree total weighted depth (absolute)}} + p_r + \underbrace{\sum_{i=r+1}^{n-1} p_i(depth_{k_b}(k_i)+1) + \sum_{i=r+1}^{n} q_i(depth_{k_r}(m_i)+1)}_{\text{right subtree total weighted depth (absolute)}}$$

Convert to "relative depth":

$$\underbrace{\sum_{i=0}^{n-1} p_i + \sum_{i=0}^{n} q_i}_{\text{total probability}} + \underbrace{\sum_{i=0}^{r-1} p_i\; depth_{k_a}(k_i) + \sum_{i=0}^{r} q_i\; depth_{k_a}(m_i)}_{\text{left subtree total weighted depth (relative)}} + \underbrace{\sum_{i=r+1}^{n-1} p_i\; depth_{k_b}(k_i) + \sum_{i=r+1}^{n} q_i\; depth_{k_r}(m_i)}_{\text{right subtree total weighted depth (relative)}}$$

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

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

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

$$q_i + C[i+1][j] \qquad\qquad C[i][r-1] + C[r+1][j] \qquad\qquad C[i][j-1] + q_{j+1}$$

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

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

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

	0	1	2	3	4	5	6	7
k_i	a	do	i	in	like	not	then	you
p_i	.073	.045	.104	.05	.055	.103	.076	.042
q_i	.001	.113	.107	.006	.001	.02	.081	.122	.001


```
                              0    7
                                3.205/2
                           1            6
                             2.916/2  2.819/5
                        2                    5
            i            2.119/2  2.567/5  2.125/5        j
                     3                          4
                       1.538/2  1.783/2  1.873/5  1.458/6
                  4                                    3
                    1.24/1  1.202/2  1.247/2  1.212/5  1.227/6
               5                                            2
                 1.08/1  .975/2  .666/2  .613/5  1.018/6  1.038/6
            6                                                   1
              .747/1  .818/1  .439/2  .216/4  .438/5  .829/6  .691/6
         7                                                          0
           .301   .485    .33    .064    .097    .305    .482    .288
```
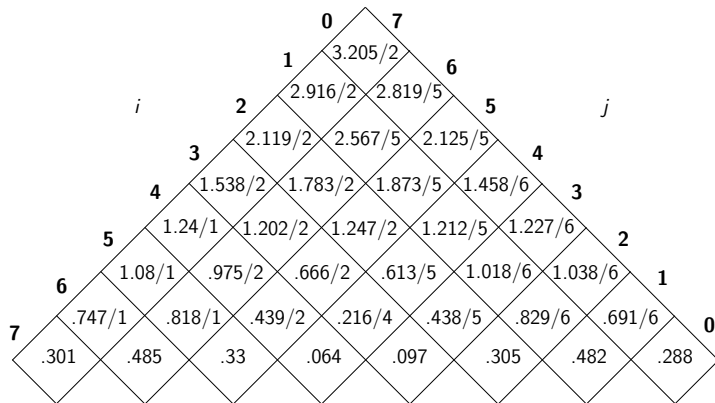
Coming up:

Due **Fri, March 31** *(end of day)*
Read Section 6.5
(No quiz on Section 6.5)

Due **Mon, Apr 3** *(end of day)*
Do Project 6.b (as practice problem, not a project)

Due **Mon, Apr 10** *(end of day)*
Read Sections 7.(1 & 2)
Take quiz

Due **Tues, Apr 11** *(end of day) Do Project 7.1 (as a practice problem, not a project)*

Do **Optimal BST** *project (suggested by Monday, Apr 10)*