**Analysis of Searching Times**

Given a binary search tree:

```
        H
       /\    /
      D --L  G
     /  /  /  /
    B  F  J  N
   /\   /\   /\    /
  A C E I K M O
```

Analyze how much time might be spent in searching, on average.

Possible assumptions:
1. All items in tree equally likely
   or
   Some items more likely than others
2. Specific items know ahead of time
   or
   Exact data cannot be anticipated
3. Data known to be in tree
   or
   Data may or may not be in tree

**Terminology**

**Level of a node**
- Many definitions may be found
  1. Number of nodes on path from given node to root
  2. Number of comparisons needed to find node, starting at root.

(Note Def 2 = 1 + Def 1)

In this class, use Definition 2
(Dr. Dale uses Definition 1)

**Internal Path Length**
- Sum of levels of nodes within tree

If all nodes equally likely in search and if data is in tree, then

\[
\text{Avg search steps} = \frac{\text{Internal path length}}{n}
\]

**External node (or failure node)**
- Object 'pointed to' by nil pointer

(Other nodes sometimes called internal nodes)

**External Path Length**
- Sum of levels of external nodes
If data not in tree and if all failure nodes equally likely in search, then

\[ \text{Avg steps} = \frac{\text{External Path Length}}{(n+1)} \]

Now, suppose we know the normal frequency that we will see each node (internal or external) during the search.

Given a tree with nodes \( a_1, \ldots, a_n \)
Let \( f_i \) be the frequency of node \( a_i \)

Then

- the \textit{weight of the tree} = \( f_1 + \ldots + f_n \)
- the \textit{weighted internal (or external) path length} is
  \[ f_1 \times \text{level}(a_1) + \ldots + f_n \times \text{level}(a_n) \]

Sometimes this is also called the \textit{weighted cost} of the tree.

**Computational Formula:**

Suppose \( T \) is a tree with
- root \( R \)
- left subtree \( T_L \)
- right subtree \( T_R \)

How does weighted cost of \( T \) relate to data we might know about \( R, T_L, \) and \( T_R \)?

First, look at example:

```
    H
   / \   / \\
  D   L  B    F
 / \ / \ / \ \\
A  C E  G I K M O
```

Note: Level of each node in each subtree increases by 1.
Now consider computation of weighted cost (WC) for overall tree:

\[ WC = f_1 \cdot \text{level}(a_1) + \ldots + f_n \cdot \text{level}(a_n) \]

where levels are in new tree

If we consider \( a_1 \) as the root of this new tree

\[ WC = f_{\text{root}} + f_2 \cdot \text{level}(a_2) + \ldots + f_n \cdot \text{level}(a_n) \]

\[ = f_{\text{root}} + f_2 \cdot (1 + \text{level}_{\text{old}}(a_2)) + \ldots + f_n \cdot (1 + \text{level}_{\text{old}}(a_n)) \]

\[ = f_{\text{root}} + f_2 + f_2 \cdot \text{level}_{\text{old}}(a_2) + \ldots + f_n + f_n \cdot \text{level}_{\text{old}}(a_n) + \]

\[ = f_{\text{root}} + f_2 + \ldots + f_n + \text{weighted cost for both subtrees} \]

\[ = \text{weight of new tree} + \text{weighted cost of subtrees} \]

---

**Notation**

Given binary search tree \( T \)
- Root \( R \)
- Subtrees \( T_L, T_R \)
- \( n \) nodes, \( a_1, \ldots, a_n \)
- \( \text{level}(N) = \# \text{nodes on path to root} \)
- \( \text{level}(R) = 1 \)
- Weights or frequencies \( f_1, \ldots, f_n \)

\[ \text{Weight}(T) = f_1 + \ldots + f_n \]

\[ \text{Weighted Cost}(T) = \sum f_i \cdot \text{level}(a_i) \]

\[ = \text{Weight}(T) + WC(T_L) + WC(T_R) \]
Problem: Put nodes in tree so weighted cost is a minimum.

Result is called an optimal binary search tree.

Observation 1:

If $T$ is an optimal search tree for $A_1, \ldots, A_n$, then so are $T_1, T_2$.

Observation 2:

How to build $T$, given $A_1, \ldots, a_n$?

One approach:

Consider each node $A_1, \ldots, a_n$.

For $A_n$,

find optimal tree for $A_1, \ldots, a_{n-1} = T_{n-1}$

find optimal tree for $A_{n-1}, \ldots, a_n = T_{n-1}$

Then compare possibilities

Choose best possibility.
Example 1

Given $a_1, a_2, a_3, a_4, a_5$

Frequencies 9 3 4 8 1

Easiest to work bottom up

Best trees - 1 node

$T_1 = a_1, T_2 = a_2, T_3 = a_3, T_4 = a_4, T_5 = a_5$

Weight 9 3 4 8 1

WC 9 3 4 8 1

root $a_1, a_2, a_3, a_4, a_5$

Best trees - 2 (adjacent) nodes

$T_{12}, T_{23}, T_{34}, T_{45}$

Weight 12 12 7 7 12 12 9 9

WC 15 21 11 16 20 16 10 17

root $a_1, a_2, a_3, a_4, a_5$

Best trees - 3 (adjacent) nodes

$T_{13}, T_{24}, T_{35}$

Weight 16 16 16 15 15 15 13 13 13

WC 26 28 31 35 26 25 26 18 29

root $a_1, a_2, a_3, a_4, a_5$

use $T_{13}, a_3, T_{12}, T_{34}, a_3, T_{23}, T_{35}$

Best trees - 4 (adjacent) nodes

$T_{14}, T_{25}$

Weight 24 24 24 24 16 16 16 16

WC 49 49 47 50 42 29 47 41

root $a_1, a_2, a_3, a_4, a_5$

use $T_{14}, a_3, T_{12}, a_4, T_{23}, T_{35}$
Best trees - 5 nodes

Next, consider possibility search can fail,

Where frequency of failure also known

Approach is similar, except failure nodes cannot be internal.

Must decide level to use for external node
- could use level of parent
- could use one level more.

Change notation to label subtrees by failure nodes.

Weighted Cost = 3.3 + 9.2 + 4.1 + 8.2 + 1.3
= 9 + 18 + 4 + 16 + 3
= 50

T15

<table>
<thead>
<tr>
<th>Weight</th>
<th>25</th>
<th>25</th>
<th>25</th>
<th>25</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>U/C</td>
<td>5-2</td>
<td>9-25</td>
<td>5-2</td>
<td>5-2</td>
<td>72</td>
</tr>
<tr>
<td>Level</td>
<td>T5</td>
<td>a1, b5</td>
<td>T15, b5</td>
<td>T15, a5</td>
<td>T15</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
T15
\end{align*}
\]

\[
\begin{align*}
\text{Weighted Cost} &= 3.3 + 9.2 + 4.1 + 8.2 + 1.3 \\
&= 9 + 18 + 4 + 16 + 3 \\
&= 50
\end{align*}
\]
Example - Counting External level = Level Parent

Node:  C  F  J  P (p's)
Frequency:  5  10  3  8 (p's)
Failure Preg:  1  2  2  1  3 (p's)

Again, work proceeds bottom-up

Trees - external node

Weight:  T0  T1  T2  T3
Weight:  1  2  2  1  3
WC:  0  0  0  0  0
Root:  -  -  -  -  -
Use:  -  -  -  -  -

Trees - 2 external nodes

Weight:  T01  T12  T23  T34
Weight:  8  0  14  6  12
WC:  0  0  0  0  0
Root:  C  F  J  P
Use:  T0, T1  T1, T2  T2, T3  T3, T4

Trees - 3 external nodes

Weight:  T02  T13  T24
Weight:  20  20  18  17
WC:  0  0  17  0
Root:  C  F  J  P
Use:  T0, T1, T2  T1, T3  T2, T3  T3, T4

Trees - 4 external nodes

Weight:  T03  T14
Weight:  24  24  24  24
WC:  0  0  0  0
Root:  C  F  J  P
Use:  T0, T1, T2, T3  T1, T3  T2, T3  T3, T4
Example - With External Level = Parent Level + 1

Node: C F J P
Frequency: 5 10 3 8 (p's)
Failure Freq.: 1 2 2 1 3 (q's)

As before, work proceeds bottom-up.
Only change from before in first step.

Trees - 1 external node

<table>
<thead>
<tr>
<th>Node</th>
<th>T00</th>
<th>T11</th>
<th>T22</th>
<th>T33</th>
<th>T44</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>WC</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Root</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Trees - 2 external nodes

<table>
<thead>
<tr>
<th>Node</th>
<th>T01</th>
<th>T12</th>
<th>T23</th>
<th>T34</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>8</td>
<td>14</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>WC</td>
<td>11</td>
<td>18</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>Root</td>
<td>C</td>
<td>F</td>
<td>J</td>
<td>P</td>
</tr>
<tr>
<td>Use</td>
<td>T00,T11</td>
<td>T12,T22</td>
<td>T23,T33</td>
<td>T34, T44</td>
</tr>
</tbody>
</table>
Trees - 3 external nodes

<table>
<thead>
<tr>
<th>Tree</th>
<th>Weight</th>
<th>WC</th>
<th>Root</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_{o2}</td>
<td>20 \times 19 + 18</td>
<td>39 \times 33</td>
<td>C</td>
<td>T_{o2}, T_{o3}</td>
</tr>
<tr>
<td>T_{i3}</td>
<td>18 \times 27 + 17 \times 37</td>
<td>21 \times 35 + 9 \times 29</td>
<td>F</td>
<td>T_{i3}, T_{i4}</td>
</tr>
<tr>
<td>T_{i4}</td>
<td>17 \times 29</td>
<td>9 \times 29</td>
<td>J</td>
<td>T_{i3}, T_{i4}</td>
</tr>
</tbody>
</table>

Trees - 5 external nodes

<table>
<thead>
<tr>
<th>Tree</th>
<th>Weight</th>
<th>WC</th>
<th>Root</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_{o4}</td>
<td>35 \times 40 + 35 \times 29</td>
<td>96 \times 75</td>
<td>C</td>
<td>T_{o4}, T_{o5}</td>
</tr>
<tr>
<td>T_{i5}</td>
<td>35 \times 84</td>
<td>35 \times 84</td>
<td>F</td>
<td>T_{i5}, T_{i6}</td>
</tr>
<tr>
<td>T_{i6}</td>
<td>25 \times 82</td>
<td>P</td>
<td>J</td>
<td>T_{i5}, T_{i6}</td>
</tr>
</tbody>
</table>

Trees - 4 external nodes

<table>
<thead>
<tr>
<th>Tree</th>
<th>Weight</th>
<th>WC</th>
<th>Root</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_{o3}</td>
<td>24 \times 54</td>
<td>54 \times 49</td>
<td>C</td>
<td>T_{o3}, T_{o4}</td>
</tr>
<tr>
<td>T_{i4}</td>
<td>24 \times 64</td>
<td>24 \times 60</td>
<td>F</td>
<td>T_{i4}, T_{i5}</td>
</tr>
<tr>
<td>T_{i5}</td>
<td>29 \times 63</td>
<td>29 \times 61</td>
<td>J</td>
<td>T_{i4}, T_{i5}</td>
</tr>
</tbody>
</table>

Diagram:

```
     F
    / \  \
   /   \ /
  /     \|
 P     C
    /\ /
   / \|
  1   3
```

```