Optimal Binary search tree

Explanation:

suppose we are searching a word from a dictionary and for every required word, we are looking in the dictionary, then it is consider as a time consuming process.

In order to perform an efficient search, we are moving to Binary search tree. By using common words as key element.

Again we are looking for frequently used words nearer to the root word and least frequently used words far away from root word. This methodology is called optimal Binary search tree (OBST).

Let $a_1 a_2 \dots a_n$ be set of identifies that $a_1 < a_2 < a_3 \dots$ Let $p(i)$ be the probability with which we can search for $a_i \Rightarrow i$

Tree should be build upon optimal cost $\sum_{i=1}^{n} p(i)$ $\sum_{i=1}^{n} q(i) \Rightarrow$ OBST.

$$c[i,j] = \min_{i \le k \le j} \left\{ c[i,k-1] + c[k+1,j] + \sum_{s=i}^{j} p_s \right\}$$

where $1 \le i \le j \le n$.

Initially we assume $c[i, i-1] = 0$ for i from 1 to $n+1$.

Then set $c[i,j] = p_i$ where $1 \le i \le n$

→ for OBST, we need to build

↳ cost table

↳ root table.

Ex:

$\{do, if, int, while\} = \{0.1, 0.2, 0.4, 0.3\}$

a[i]	1	2	3	4
p[i]	0.1	0.2	0.4	0.3

root table

cost table

j -> 0 to n

i -> 1 to n+1

	0	1	2	3	4
1	0	0.1			
2		0	0.2		
3			0	0.4	
4				0	0.3
5					0

	1	2	3	4
1	1			
2		2		
3			3	
4				4

we need to find $c(1,2), c(1,3), c(1,4), c(2,3), c(2,4)$ and $c(3,4)$

$$c[i,j] = c(i, k-1] + c[k+1, j] + \sum_{s=i}^{j} p_s$$

$c(1,2)$ $i=1$ $j=2$ $k=1$ and 2, we have to find for $k=1$ and $k=2$

if 1,3 we have to find for $k=1,2,3$.

$c(1,2) = \boxed{\text{if } k=1}$

$$c[1,2] = c(1,0) + c(2,2) + \sum_{1}^{2} p(1)+p(2)$$

$$= c(1,0) + c(2,2) + p(1) + p(2)$$

$$= 0 + 0.2 + 0.1 + 0.2$$

$c[1,2] = 0.5$

if $\boxed{k=2}$

$$c[1,2] = c[1,1] + c[3,2) + p(1) + p(2)$$

$$= 0.1 + 0 + 0.1 + 0.2$$

$c(1,2) = \underline{\underline{0.4}}$ -> This is min for $k=2$.

hence cost table

	0	1	2	3	4
1	0	0.1	0.4		
2		0	0.2		
3			0	0.4	
4				0	0.3
5					0

root table

	1	2	3	4
1	1	2		
2		2		
3			3	
4				4

$c(2,3)$ $i=2$ $j=3$ $k=2$ and 3

if $\boxed{k=2}$

$c(2,3) = c(2,1) + c(3,3) + p(2) + p(3)$

$= 0 + 0.4 + 0.2 + 0.4$

$c(2,3) = 1$

$\boxed{k=3}$

$c(2,3) = c(2,2) + c(4,3) + p(2) + p(3)$

$= 0.2 + 0 + 0.2 + 0.4$

$c(2,3) = 0.8 \rightarrow$ min

hence cost table

	0	1	2	3	4
1	0	0.1	0.4		
2		0	0.2	0.8	
3			0	0.4	
4				0	0.3
5					0

root table

	1	2	3	4
1	1	2		
2		2	3	
3			3	
4				4

$c(3,4) =$ $i=3$ $j=4$ $k=3,4$

$\boxed{k=3}$

$c(3,4) = c(3,3) + c(3,1) + p(3) + p(4)$

$= 0 + 0.3 + 0.4 + 0.3$

$c(3,4) = 1.0 \rightarrow$ This is min.

If $k=4$.

$c(3,4) = c(3,3) + c(5,4) + p(3) + p(4)$

$= 0.4 + 0 + 0.4 + 0.3 \Rightarrow 1.1$

	0	1	2	3	4
1	0	0.1	0.4		
2		0	0.2	0.8	
3			0	0.4	1.0
4				0	0.3
5					0

	1	2	3	4
1	1	2		
2		2	3	
3			3	3
4				4

c(1,3) k=1,2,3 i=1 j=3

if $\boxed{k=1}$

c(1,3) = c(1,0) + c(2,3) + p(1) + p(2) + p(3)

= 0 + 0.8 + 0.1 + 0.2 + 0.4

c(1,3) = 1.5

$\boxed{k=2}$

c(1,3) = c(1,1) + c(3,3) + p(1) + p(2) + p(3)

= 0.1 + 0.4 + 0.1 + 0.2 + 0.4

c(1,3) = 1.2

$\boxed{k=3}$

c(1,3) = c(1,2) + c(4,3) + p(1) + p(2) + p(3)

= 0.4 + 0 + 0.1 + 0.2 + 0.4

c(1,3) = 1.1 -> This is min.

cost table

	0	1	2	3	4
1	0	0.1	0.4	1.1	
2		0	0.2	0.8	
3			0	0.4	1.0
4				0	0.3
5					0

root table

	1	2	3	4
1	1	2	3	
2		2	3	
3			3	3
4				4

c(2,4) i=2 j=4 k=2, 3, 4.

$\boxed{k=2}$

c(2,4) = c[c(2,1) + c(3,4) + p(2) + p(3) + p(4)

= 0 + 1.0 + 0.9 + 0.4 + 0.3 + 0.2

c(2,4) = 1.9

$\boxed{k=3}$

$c(2,4) = c(2,2) + c(4,4) + p(2) + p(3) + p(4)$

$= 0.2 + 0.3 + 0.2 + 0.4 + 0.3$

$c(2,4) = \underline{\underline{1.4}}$ → This is min.

$\boxed{K=4}$

$c(2,4) = c(2,3) + c(5,4) + p(2) + p(3) + p(4)$

$= 0.8 + 0 + 0.2 + 0.4 + 0.3$

$c(2,4) = \underline{1.7}$

and last $c(1,4)$ $i=1$ $j=4$ $k=1, 2, 3, 4.$

$\boxed{K=1}$

$c(1,4) = c(1,0,) + c(2,4) + p(1) + p(2) + p(3) + p(4)$

$= 0 + 1.4 + 0.1 + 0.2 + 0.4 + 0.3$

$= \underline{\underline{2.4}}.$

$\boxed{K=2}$

$c(1,4) = c(1,1) + c(3,4) + p(1) + p(2) + p(3) + p(4)$

$= (0.1 + 1.0 + 1.0 +$ since $(0.1 + 0.2 + 0.4 + 0.3 = 1)$

$c(1,4) = \underline{\underline{2.1}}$

$\boxed{K=3}$

$c(1,4) = c(1,2) + c(4,4) + \sum_{s=1}^{4} p(s)$

$= 0.4 + 0.3 + 1.0$

$c(1,4) = \underline{\underline{1.7}}$ → This is min

$\boxed{K=4}$

$c(1,4) = c(1,3) + c(5,4) + \sum_{s=1}^{4} p(s)$

$c(1,4) = 1.1 + 0 + 1.0 \Rightarrow \underline{2.1}$

hence cost table

	0	1	2	3	4
1	0	0.1	0.4	1.1	1.7
2		0	0.2	0.8	1.4
3			0	0.4	1.0
4				0	0.3
5					0

root table.

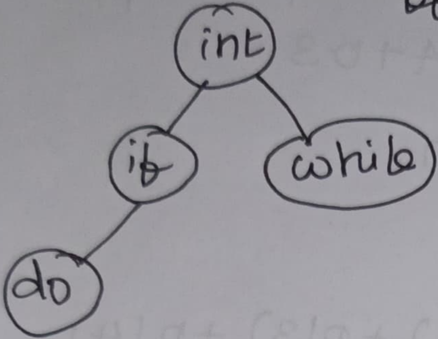
	1	2	3	4
1	1	2	3	3
2		2	3	3
3			3	3
4				4

In root table 1 to 3 → 3 is higher probability.

for 3 → it is 'int' keyword.

we have do, if, int, while

left right



above is OBST order for given key word.

OBST Algo: Continue.....

cost table c[i, j]

Root table R[i, j]

```
for(i<-1 to n) do
{
   c[i, i-1] <- 0
   c[i, i] <- p(i)
   R[i, i] <- i
}
```
n times execution.

```
c[n+1, n] <- 0
for(m<-1 to n-1) do
{
 for(i<-1 to n-3) do
 {
 j <- i+m
 min <- ∞
```
n times execution

```
 for(k<-i to j) do
 {
 if((c[i,k-1] + c[k+1, j]) < min) then
 {
 min <- c[i, k-1] + c[k+1, j]
 k_min <- k;
 }
 }
```
n times execution

```
 R[i,j] <- k
 Sum <- p[i]
 for s <- i+1 to j do
 sum = sum + p[s]
   c[i,j] = min + sum
 }
}
for(i=1 to n+1) do
 for(j=0 to n) do
 {
   write c[i][j])
 }
```

```
for (i ← 1 to n) do
{
  for (j ← 1 to n) do
  {
    write R[i][j]
  }
}
```

$$C(n) = \sum_{m=1}^{n-1} \sum_{i=1}^{n-m} \sum_{k=i}^{j} 1$$

$$= n^3$$

$\boxed{C(n) = \Theta(n^3)}$ → time complexity of optimal Binary search tree algorithm.