Design and Analysis of Algorithms

Lecture – 22

Dynamic Programming – Optimal Binary Search Tree

Success is always inevitable with Hard Work and Perseverance

N. Ravitha Rajalakshmi

Learning Objective

- Derive optimal solution for organizing set of keys in a binary search tree such that the cost of the search is reduced.
- (Only solutions using Tabulation are discussed)

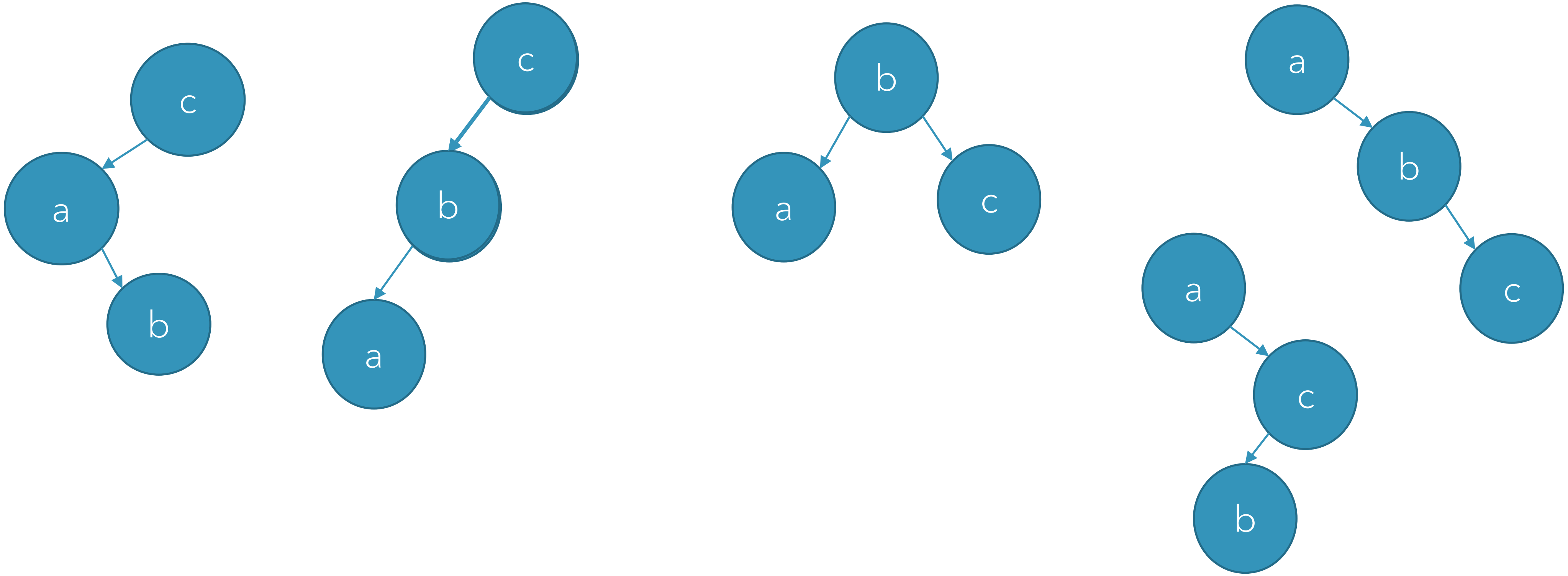
Optimal Binary Search Tree

- Given a set of keys {c, a, b} along with their probability of occurrence{0.4, 0.4 0.2} arrange the keys in BST such that average cost of search for the keys is reduced.

- BST a special tree where a node can have at most two children and keys are ordered

Optimal Binary Search Tree

- What are the different arrangements possible with keys c, a and b

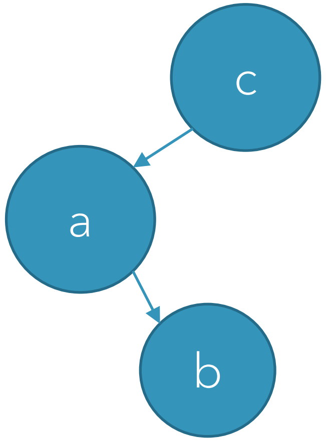


Cost of the tree

- Let n_i denote number of search for key i in the tree and P_i denote the probability of occurrence of key i

cost of the tree = $\sum_{i=1}^{k} n_i P_i$

Here, k denote the number of keys



Keys	a	b	c
p_i	0.4	0.2	0.4
n_i	2	3	1

Cost = 0.4 * 2 + 0.2 *3 + 0.4 *1
= 0.8 + 0.6 + 0.4 = 1.8

Problem Instance

- Problem instance is defined by a set of keys
- Subproblem corresponds to smaller set of keys

- Smallest subproblem is constructing a tree with a single key

  Cost of the tree = Probability of the key

Optimal Substructure Property

- If the cost of the tree to be reduced then cost of the left subtree and right subtree should be optimal as well
- Knowing the cost of left and right subtree , adding a new key as a root node will increase the cost of keys in the subtrees by one.

$$\text{cost}[i, j] = \min_{i \le k \le j} \left(\text{cost}[i, k-1] + \text{cost}[k+1, j] + \sum_{l=i}^{j} p_l \right)$$

if key k is root node, Left subtree – keys from i to k-1 and Right subtree – keys from k+1 to j

Pause & Think

- What does cost [i,j] with i>j indicate?

  Indicates the empty tree

- Why probability value of all keys is added to cost?

  All the keys in left subtree and right subtree will have their search cost increased by one

Problem

- Order the keys based on their value and number them from 1 to k

Keys	a	b	c
p_i	0.4	0.2	0.4

Create a main table and root table with rows equal to 1 to k and columns from 0 to k

Problem

Keys	a	b	c
p_i	0.4	0.2	0.4

Main Table

i/j	0	1	2	3
1	0			
2	0	0		
3	0	0	0	

Root Table

i/j	0	1	2	3
1	-			
2	-	-		
3	-	-	-	

Shaded portions correspond to empty tree whose cost would be zero
Only for the remaining cells cost needs to be computed

Problem

Keys	a	b	c
p_i	0.4	0.2	0.4

First find cost of trees with single key

Cost = probability of key
Root should be key value

Main Table

i/j	0	1	2	3
1	0	0.4		
2	0	0	0.2	
3	0	0	0	0.4

Root Table

i/j	0	1	2	3
1	-	1		
2	-	-	2	
3	-	-	-	3

Problem

Keys	a	b	c
p_i	0.4	0.2	0.4

Find cost of trees with two keys

Use the $\text{cost}[i, j] = \min\limits_{i \le k \le j} (\text{cost}[i, k-1] + \text{cost}[k+1, j] + \sum_{l=i}^{j} p_l)$

Main Table

i/j	0	1	2	3
1	0	0.4	?	
2	0	0	0.2	
3	0	0	0	0.4

Root Table

i/j	0	1	2	3
1	-	1		
2	-	-	2	
3	-	-	-	3

Problem

Cost [1, 2] === (root = 1)

Cost[1,0] + Cost[2,2] + (0.4 + 0.2)

0 + 0.2 + 0.6 = 0.8

(root = 2)

Cost[1,1] + Cost[3,2] + (0.4+0.2)

0.4 + 0 + 0.6 = 1

Cost [1,2] = 0.8 (root as 1)

Problem

Keys	a	b	c
p_i	0.4	0.2	0.4

Find cost of trees with two keys

Use the

$$\text{cost}[i\,,j] = \min_{i \le k \le j} \left(\text{cost}[\,i, k-1\,] + \text{cost}[k+1, j] + \sum_{l=i}^{j} p_l \right)$$

Main Table

i/j	0	1	2	3
1	0	0.4	0.8	
2	0	0	0.2	?
3	0	0	0	0.4

Root Table

i/j	0	1	2	3
1	-	1	1	
2	-	-	2	
3	-	-	-	3

Problem

Cost [2, 3]   === (root = 2)

Cost[2,1] + Cost[3,3] + (0.2 + 0.4)

0 + 0.4 + 0.6 = 1.0

(root = 3)

Cost[2,2] + Cost[4,3] + (0.2+0.4)

0.2 + 0 + 0.6 = 0.8

Cost [2,3] = 0.8 (root as 3)

Problem

Keys	a	b	c
p_i	0.4	0.2	0.4

Find cost of trees with three keys

Use the $\text{cost}[i, j] = \min\limits_{i \le k \le j} (\text{cost}[i, k-1] + \text{cost}[k+1, j] + \sum_{l=i}^{j} p_l)$

Main Table

i/j	0	1	2	3
1	0	0.4	0.8	?
2	0	0	0.2	0.8
3	0	0	0	0.4

Root Table

i/j	0	1	2	3
1	-	1	1	
2	-	-	2	3
3	-	-	-	3

Problem

Cost [1, 3] === (root = 1)

Cost[1,0] + Cost[2,3] + (0.4 + 0.2 + 0.4)

0 + 0.8 + 1.0 = 1.8

(root = 2)

Cost[1,1] + Cost[3,3] + (0.4 + 0.2 + 0.4)

0.4 + 0.4 + 1.0 = 1.8

(root = 3)

Cost[1,2] + Cost[4,3] + (0.4 + 0.2 + 0.4)

0.8 + 0 + 1.0 = 1.8

Cost [1,3] = 1.8 (root as 1 (or) 2 (or) 3)

Problem

Keys	a	b	c
p_i	0.4	0.2	0.4

Find cost of trees with three keys

Use the $cost[i\ ,\ j] = \min\limits_{i \le k \le j} \left(cost\ [\ i,\ k\text{-}1\ ] + cost[k+1,j]\ + \sum_{l=i}^{j} p_l \right)$

Main Table

i/j	0	1	2	3
1	0	0.4	0.8	1.8
2	0	0	0.2	0.8
3	0	0	0	0.4

Root Table

i/j	0	1	2	3
1	-	1	1	1,2,3
2	-	-	2	3
3	-	-	-	3

Optimal Tree Construction

- Lookup on the root table for keys from 1 to 3

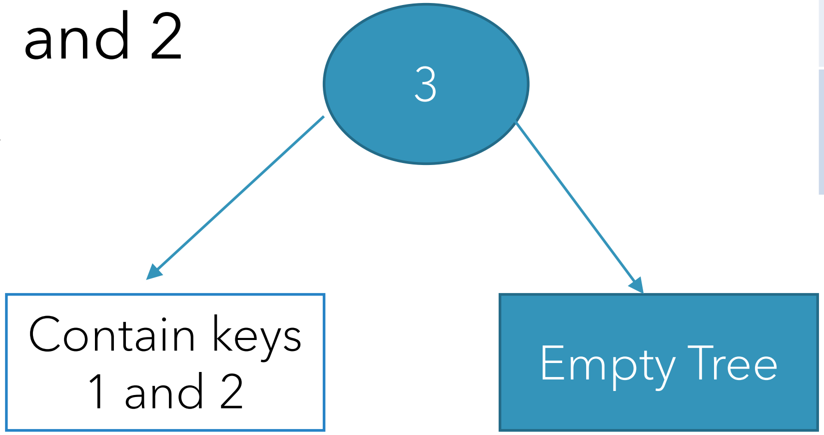
Here there are three possibilities

If we select 3 as root node

Left subtree contains 1 and 2

Right subtree is empty

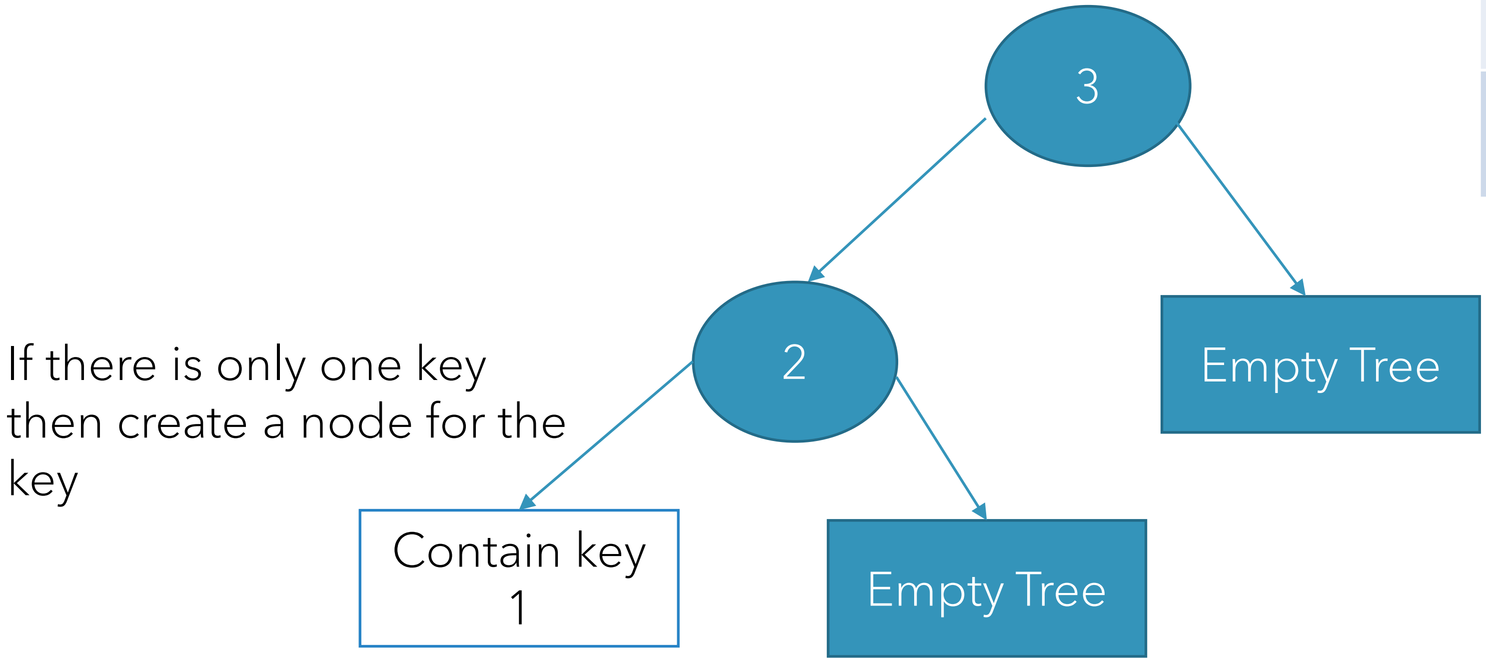
i/j	0	1	2	3
1	-	1	1	1,2,3
2	-	-	2	3
3	-	-	-	3



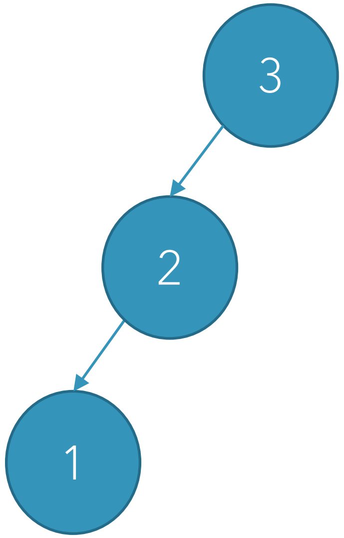
Optimal Tree Construction

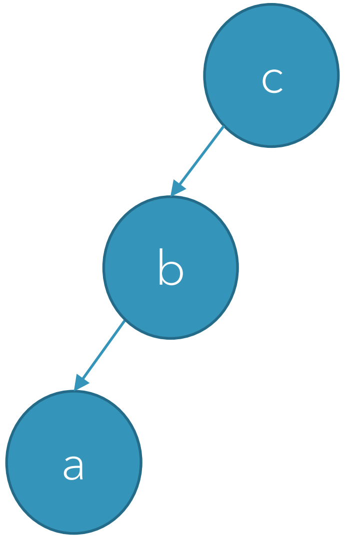
- Lookup on Cell [1,2] , the value is 2
- So 2 will be the root node in that subtree

i/j	0	1	2	3
1	-	1	1	1,2,3
2	-	-	2	3
3	-	-	-	3



Optimal Tree





Summary

- Discussed about dynamic programming solution for OBST (Optimal Binary Search Tree).

Thank You
Happy Learning

Success is always inevitable with Hard Work and Perseverance