Design and Analysis of Algorithms

Mohammad GANJTABESH

mgtabesh@ut.ac.ir

School of Mathematics, Statistics and Computer Science,
University of Tehran,
Tehran, Iran.

Techniques for the design of Algorithms

The classical techniques are as follows:

1. Divide and Conquer
2. Dynamic Programming
3. Greedy Algorithms
4. Backtracking Algorithms
5. Branch and Bound Algorithms

Optimal Binary Tree Construction

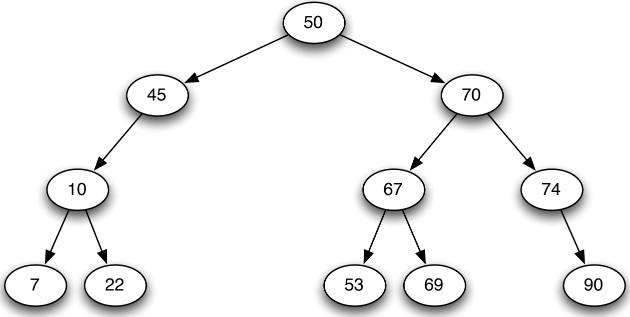
Properties

- $\forall Node \in BST.Nodes$,
  $Key(Node.Left) < Key(Node) < Key(Node.Right)$.
- Inorder traversal of any BST is sorted.
- $\log(n) \leq Depth(BST) \leq n$.

Optimal Binary Tree Construction

Properties

- $\forall Node \in BST.Nodes$,
  $Key(Node.Left) < Key(Node) < Key(Node.Right)$.
- Inorder traversal of any BST is sorted.
- $\log(n) \leq Depth(BST) \leq n$.



Optimal Binary Tree Construction

Optimal Binary Tree Construction

Given an array of n sorted integers and a searching probability for each element, i.e.:

Integers	x_1	x_2	$\cdots$	x_n
Probabilities	p_1	p_2	$\cdots$	p_n

, where $\sum_{i=1}^{n} p_i = 1$. Construct a Binary Search Tree in such a way that minimize

$$Cost = \sum_{i=1}^{n} p_i \times (Depth(x_i) + 1).$$

Optimal Binary Tree Construction

Optimal Binary Tree Construction

Given an array of n sorted integers and a searching probability for each element, i.e.:

Integers	x_1	x_2	$\cdots$	x_n
Probabilities	p_1	p_2	$\cdots$	p_n

, where $\sum_{i=1}^{n} p_i = 1$. Construct a Binary Search Tree in such a way that minimize

$$Cost = \sum_{i=1}^{n} p_i \times (Depth(x_i) + 1).$$

Recall that the number of different binary trees with exactly n node can be expressed by:

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

Example

Example

Consider the following instance:

Integers	1	2	3	4	5	6	7	8
Probabilities	0.2	0.1	0.3	0.07	0.15	0.04	0.08	0.06

Two possible Binary Search Trees are as follows:

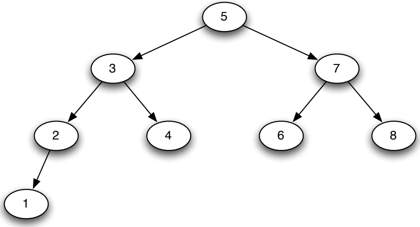
Example

Example

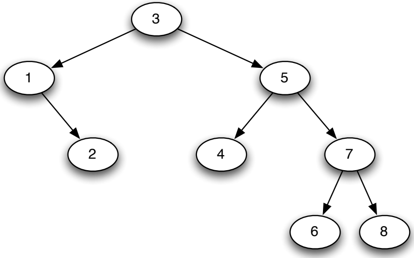
Consider the following instance:

Integers	1	2	3	4	5	6	7	8
Probabilities	0.2	0.1	0.3	0.07	0.15	0.04	0.08	0.06

Two possible Binary Search Trees are as follows:



$Cost = 2.52$



$Cost = 2.15$

Optimal Binary Tree Construction

Solution

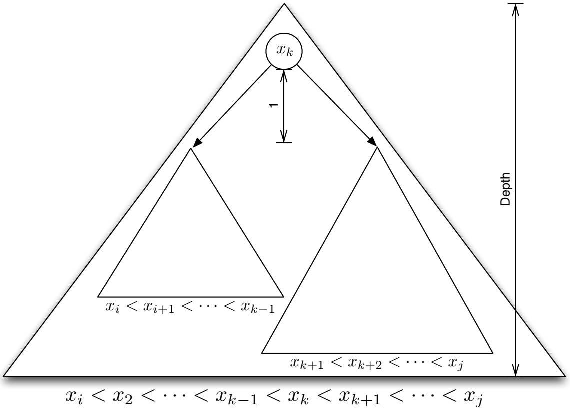
Let $C_{i,j}$ be the minimum cost for the elements $x_i, x_{i+1}, \cdots, x_j$ as follows:

$$C_{i,j} = \sum_{t=i}^{j} p_t \times (Depth(x_t) + 1).$$

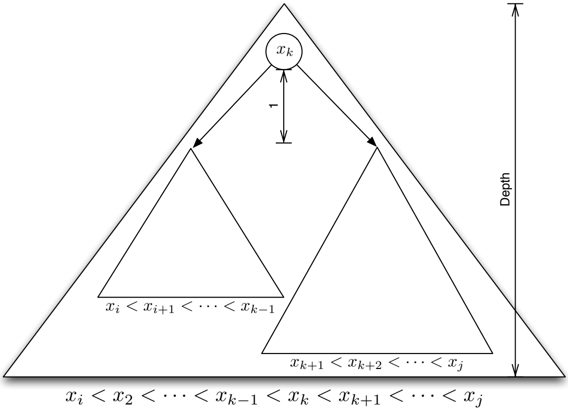
Also let $p_{i,j} = \sum_{t=i}^{j} p_t$.
Now, $C_{i,j}$ can be computed as follows:

$$C_{i,j} = \begin{cases} 0 & \text{if } i > j, \\ p_i & \text{if } i = j, \\ min_{i \le k \le j}\{C_{i,k-1} + C_{k+1,j} + p_{i,j}\} & \text{if } i < j. \end{cases}$$

Optimal Binary Tree Construction



Optimal Binary Tree Construction



$$\{C_{i,k-1} + p_{i,k-1}\} + \{C_{k+1,j} + p_{k+1,j}\} + p_k = C_{i,k-1} + C_{k+1,j} + p_{i,j}$$

Example

Example

Consider the following instance:

Integers	1	2	3	4	5
Probabilities	0.3	0.05	0.08	0.45	0.12

	1	2	3	4	5
1	0.3	0.4	0.61	1.49	1.73
2	0	0.05	0.18	0.76	1
3	0	0	0.08	0.6	0.86
4	0	0	0	0.45	0.69
5	0	0	0	0	0.12

C

	1	2	3	4	5
1	0	2	1	4	4
2	0	0	3	4	4
3	0	0	0	3	4
4	0	0	0	0	4
5	0	0	0	0	0

K

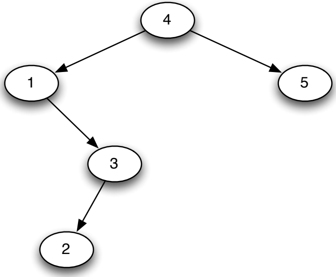
Example

	1	2	3	4	5
1	0.3	0.4	0.61	1.49	1.73
2	0	0.05	0.18	0.76	1
3	0	0	0.08	0.6	0.86
4	0	0	0	0.45	0.69
5	0	0	0	0	0.12

C

	1	2	3	4	5
1	0	2	1	4	4
2	0	0	3	4	4
3	0	0	0	3	4
4	0	0	0	0	4
5	0	0	0	0	0

K



Exercises

1. Determine the cost and structure of an optimal binary search tree for a set of $n = 6$ keys with the following probabilities:

Integers	1	2	3	4	5	6
Probabilities	0.01	0.02	0.04	0.08	0.16	0.69

2. Knuth has shown that there are always roots of optimal subtrees such that $root[i, j-1] \leq root[i, j] \leq root[i+1, j]$ for all $1 \leq i < j \leq n$. Use this fact to modify the OPTIMAL-BST procedure to run in $\Theta(n^2)$ time.