Optimal BSTs

Wednesday, 30 August 2023 12:40 PM

A binary search tree is a data structure for storing & searching for keys in a database

Keys from a total order

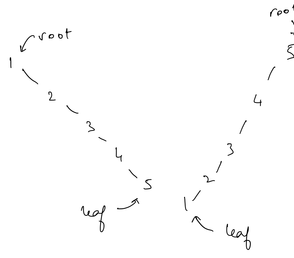
Given n keys $a_1 < a_2 < \dots < a_n$

BST stores these keys as nodes of a binary tree where for a node v,

all keys in left subtree of v $<$ key at v $<$ all keys in right subtree of v

Examples: keys are 1, 2, 3, 4, 5

Possible BSTs :



or :

(Tree terminology : root, child, parent, leaf, vertex / node, ancestor, descendant, height of a node)

Problem: Given n keys $a_1 < a_2 < \dots < a_{n-1} < a_n$

Prob of access $p_1 \quad p_2 \quad \dots \quad p_{n-1} \quad p_n$

Given a tree T, $\text{cost}_T = \sum_{i=1}^{n} p_i \, \text{height}_T(a_i)$

(assume root is at height 1)

Find a BST of minimum cost

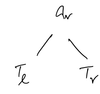
Will use dynamic programming

① Optimal Substructure :

Say a_r is root of optimal BST.

Then : $a_1 \dots a_{r-1}$ are in T_L

$a_{r+1} \dots a_n$ are in T_R



AND: T_L is optimal BST for $a_1 \dots a_{r-1}$

T_R is " " for $a_{r+1} \dots a_n$

$$\text{cost}(T) = p_r + \sum_{i<r} p_i (1 + ht_{T_L}(a_i)) + \sum_{i>r} p_i (1 + ht_{T_R}(a_i))$$

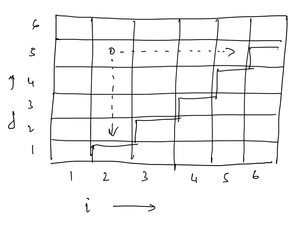
$$= \sum_{i=1}^{n} p_i + \underbrace{\sum_{i<r} p_i \, ht_{T_L}(a_i)}_{\text{cost}(T_L)} + \underbrace{\sum_{i>r} p_i \, ht_{T_R}(a_i)}_{\text{cost}(T_R)} \quad \text{—— (*)}$$

② Recurrence : $A(i,j)$ = optimal cost of BST for elts $i \dots j$ (assume $i \le j$)

$$A(i,j) = \sum_{k=i}^{j} p_k + \min_{i \le k \le j} \{ A(i,k-1) + A((k+1),j) \}$$

(from (*))

③ Table :



Time complexity is $O(n^3)$

PROBLEM: Write down algo yourself.