Design and Analysis of Algorithm

Lecture-16:
Dynamic Programming

Contents

Optimal Binary Search Tree

Problem

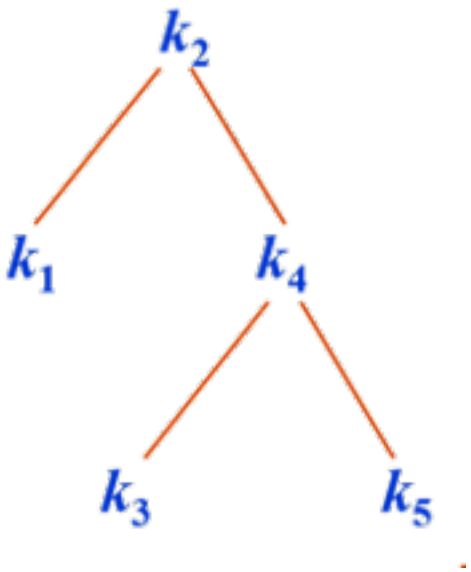
Given a sorted array keys $[0..n-1]$ of search keys and an array freq $[0..n-1]$ of frequency counts, where $freq[i]$ is the number of searches to $keys[i]$. Construct a binary search tree of all keys such that the total cost of all the searches is as small as possible.

Goal: Build a binary search tree (BST) with minimum expected search cost.

$$E[\text{search cost in } T] = \sum_{i=1}^{n} (\text{depth}_T(k_i) \cdot p_i$$

Example of search cost

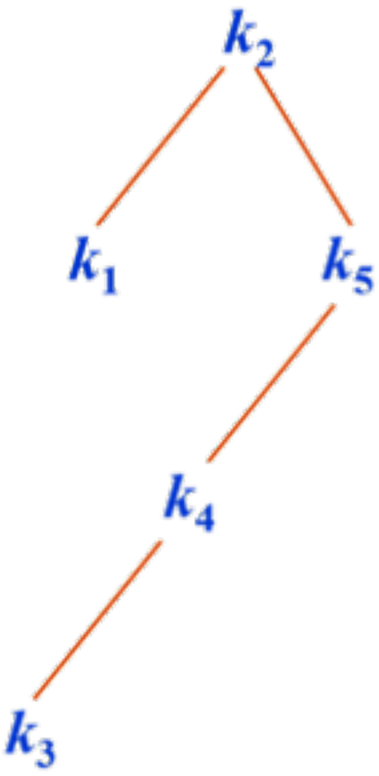
Consider 5 keys with these search probabilities: $p_1 = 0.25, p_2 = 0.2, p_3 = 0.05, p_4 = 0.2, p_5 = 0.3$.



i	$depth_T(k_i)$	$depth_T(k_i) \cdot pi$
1	2	0.5
2	1	0.2
3	3	0.15
4	2	0.4
5	3	0.9
		2.15

Example of search cost

Consider 5 keys with these search probabilities: $p_1 = 0.25, p_2 = 0.2, p_3 = 0.05, p_4 = 0.2, p_5 = 0.3$.

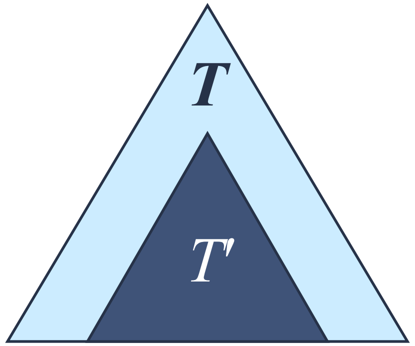


i	$depth_T(k_i)$	$depth_T(k_i) \cdot pi$
1	2	0.5
2	1	0.2
3	4	0.2
4	3	0.6
5	2	0.6
		2.1

Observation

- Optimal BST may not have smallest height.
- Optimal BST may not have highest-probability key at root.

Optimal Substructure



If T is an optimal BST and

T contains subtree T' with keys $k_i, \ldots, kj$,

then T' must be an optimal BST for keys $k_i, \ldots, kj$.

Example

Construct an optimal binary search tree over five key values k1 < k2 < k3 < k4 < k5 with access probability $0.3, 0.2, 0.1, 0.15, and\ 0.25$, respectively.

Start by considering the length of elements in tree from 1 to total number of elements in the tree

Assume that we are having tree of length 1 then

	j=1	2	3	4	5
i=1	0.3				
2		0.2			
3			0.1		
4				0.15	
5					0.25

Example

Construct an optimal binary search tree over five key values k1 < k2 < k3 < k4 < k5 with access probability $0.3, 0.2, 0.1, 0.15, \mathit{and}\ 0.25$, respectively.

	j=1	2	3	4	5
i=1	0.3	0.7^{k_1}			
2		0.2			
3			0.1		
4				0.15	
5					0.25

Assume that we are having tree of length 2 then

k1 < k2 < k3 < k4 < k5

$$P = \sum_{i=1}^{k} p_i + \min\{\mathit{possible\ options\ as\ per\ root\ selection}\}$$

$$P = .5 + \min\begin{cases} 0.2 & \mathit{if}\ k_1\ \mathit{is\ root} \\ 0.3 & \mathit{if}\ k_2\ \mathit{is\ a\ root} \end{cases}$$

Example

Construct an optimal binary search tree over five key values k1 < k2 < k3 < k4 < k5 with access probability $0.3, 0.2, 0.1, 0.15, \textit{and}\ 0.25$, respectively.

	j=1	2	3	4	5
i=1	0.3	0.7^{k_1}	1.0^{k_1}		
2		0.2	0.4^{k_2}		
3			0.1	0.35^{k_4}	
4				0.15	0.55^{k_5}
5					0.25

Assume that we are having tree of length 3 then

k1 < k2 < k3 < k4 < k5

$$P = \sum_{i=1}^{k} p_i + \min\{possible\ options\ as\ per\ root\ selection\}$$

$$P = .6 + \min\begin{cases} 0.4 & if\ k_1\ is\ root \\ 0.3 + 0.1 & if\ k_2\ is\ a\ root \\ 0.7 & if\ k_3\ is\ a\ root \end{cases}$$

Example

Construct an optimal binary search tree over five key values k1 < k2 < k3 < k4 < k5 with access probability $0.3, 0.2, 0.1, 0.15, \textit{and } 0.25$, respectively.

	j=1	2	3	4	5
i=1	0.3	0.7^{k_1}	1.0^{k_1}	1.4^{k_2}	
2		0.2	0.4^{k_2}	0.8^{k_2}	1.35^{k_4}
3			0.1	0.35^{k_4}	0.85^{k_5}
4				0.15	0.55^{k_5}
5					0.25

Assume that we are having tree of length 4 then

k1 < k2 < k3 < k4 < k5 k1 < k2 < k3 < k4 < k5

$$P=.75+\min\begin{cases} 0.8 & if\ k_1\ is\ root \\ 0.3+0.35 & if\ k_2\ is\ a\ root \\ 0.7+0.15 & if\ k_3\ is\ a\ root \\ 1.0 & if\ k_4\ is\ a\ root \end{cases}$$

$$P=.7+\min\begin{cases} 0.75 & if\ k_2\ is\ root \\ 0.2+0.55 & if\ k_3\ is\ a\ root \\ 0.4+0.25 & if\ k_4\ is\ a\ root \\ 0.8 & if\ k_5\ is\ a\ root \end{cases}$$

Example

Construct an optimal binary search tree over five key values k1 < k2 < k3 < k4 < k5 with access probability $0.3, 0.2, 0.1, 0.15, \textit{and } 0.25$, respectively.

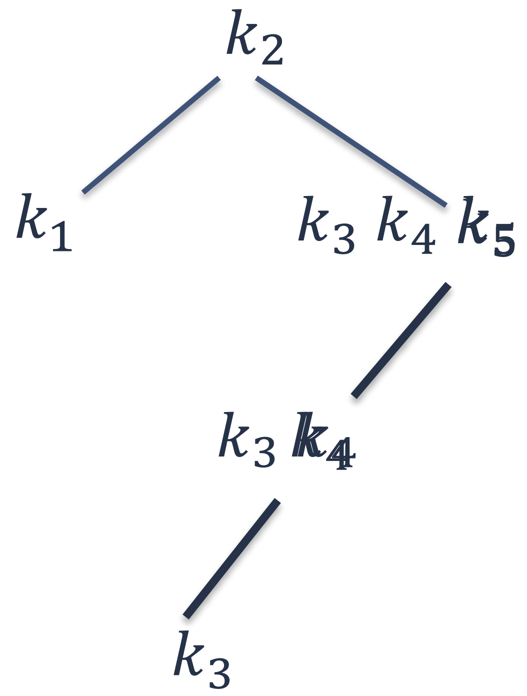
	j=1	2	3	4	5
i=1	0.3	0.7^{k_1}	1.0^{k_1}	1.4^{k_2}	2.15^{k_2}
2		0.2	0.4^{k_2}	0.9^{k_2}	1.35^{k_4}
3			0.1	0.35^{k_4}	0.85^{k_5}
4				0.15	0.55^{k_5}
5					0.25

Assume that we are having tree of length 5 then

k1 < k2 < k3 < k4 < k5

$$P=1.0+\min\begin{cases} 1.35 & \textit{if } k_1 \textit{ is root} \\ 0.3+0.85 & \textit{if } k_2 \textit{ is a root} \\ 0.7+0.55 & \textit{if } k_3 \textit{ is a root} \\ 1.0+0.25 & \textit{if } k_4 \textit{ is a root} \\ 1.4 & \textit{if } k_5 \textit{ is a root} \end{cases}$$

Example



	j=1	2	3	4	5
i=1	0.3	0.7^{k_1}	1.0^{k_1}	1.4^{k_2}	2.15^{k_2}
2		0.2	0.4^{k_2}	0.9^{k_2}	1.35^{k_4}
3			0.1	0.35^{k_4}	0.85^{k_5}
4				0.15	0.55^{k_5}
5					0.25

$$= 0.3 * 2 + 0.2 * 1 + 4 * 0.1 + 3 * 0.15 + 2 * 0.25 = 2.15$$

Question

Key	10	12	16	21
Freq.	4	2	6	3

	j=1	2	3	4
i=1	4	8^{k_1}	20^{k_3}	26^{k_3}
2		2	10^{k_3}	16^{k_3}
3			6	12^{k_3}
4				3