Algorithms Design and Analysis [ETCS-301]

Dr. A K Yadav
Amity School of Engineering and Technology
(affiliated to GGSIPU, Delhi)
akyadav1@amity.edu
akyadav@akyadav.in
www.akyadav.in
+91 9911375598

September 2, 2019

Optimal binary search trees I

- Suppose we want to store English dictionary
- We can use any balanced binary search try to store the dictionary word
- If every word of dictionary is to be searched same number of times
- we can search any world in $O(\lg n)$
- Total time taken is $O(mn \lg n)$
  where $m =$ number of times each word to be searched
  $n =$ number of words
- This system works perfectly if every word is searched equal number of times.
- But if the some words are searched very frequently and some are very rare then this performance of this system goes slow
- An alternate of this is Optimal binary search trees

Optimal binary search trees II

- Let a $K = \langle k_1, k_2, \ldots k_n \rangle$ sequence of n distinct key in the sorted order such that $k_1 < k_2 < \cdots < k_n$
- If we search keys not in database that we represent as dummy keys d_i
- Total key and dummy keys will be $d_0 < k_1 < d_1 < k_2 \ldots d_{n-1} < k_n < d_n$
- d_0 represents all keys less than k_1 and not in database, so are dummy keys
- d_n represents all keys greater then k_n and not in database, so are dummy keys
- d_i for $0 < i < n$ represents all keys greater then k_i and less than k_{i+1} and not in database, so are dummy keys.
- So Total no of n Keys are $k_1, k_2, \ldots k_n$
- So Total no of $n + 1$ Dummy Keys are $d_0, d_1, d_2, \ldots d_n$

Optimal binary search trees III

- Let probability of key k_i to be searched is p_i for $1 \leq i \leq n$.
- Let probability of dummy key d_i to be searched is q_i for $0 \leq i \leq n$.
- If we build some binary search tree for these keys and dummy keys then keys will be on internal nodes and represents successful search and dummy keys will be on external nodes and represents unsuccessful search.
- Every search is either successful finding some key k_i or unsuccessful finding some dummy key d_i so:

$$\sum_{k=1}^{n} p_k + \sum_{k=0}^{n} q_k = 1$$

Optimal binary search trees IV

- if k_r is the root of the tree then two subtree will be: one subtree having keys from k_1 to k_{r-1} and dummy keys d_0 to d_{r-1} and other subtree having keys from k_{r+1} to k_n and dummy keys d_r to d_n
- if we search any keys from k_i to k_j then dummy keys will be d_{i-1} to d_j for $1 \leq i \leq j \leq n$
- If $j = i - 1$ then there is no key but only one dummy key d_{i-1}
- Let

$$w(i,j) = \sum_{k=i}^{j} p_k + \sum_{k=i-1}^{j} q_k$$

Optimal binary search trees V

- k_r is the root of the tree having keys from k_i to k_j and dummy keys from d_{i-1} to d_j then two subtree will be: one subtree having keys from k_i to k_{r-1} and dummy keys d_{i-1} to d_{r-1} and other subtree having keys from k_{r+1} to k_j and dummy keys d_r to d_j
- The actual cost of a search equals the number of nodes examined i.e. the depth of the node found by the search in tree plus 1
- Then the expected cost of a search in tree for al keys and dummy keys is

$$E[1,n] = \sum_{k=1}^{n} (depth(k_k) + 1) . p_k + \sum_{i=0}^{n} (depth(d_k) + 1) . q_k$$

$$E[1,n] = \sum_{k=1}^{n} depth(k_k) . p_k + \sum_{i=0}^{n} depth(d_k) . q_k + \sum_{k=1}^{n} p_k + \sum_{k=0}^{n} q_k$$

Optimal binary search trees VI

$$E[1,n] = \sum_{k=1}^{n} depth(k_k).p_k + \sum_{i=0}^{n} depth(d_k).q_k + 1$$

- The expected cost of a search in tree for keys from k_i to k_j and dummy keys d_{i-1} to d_j is

$$E[i,j] = \sum_{k=i}^{j} \left(depth(k_k) + 1\right).p_k + \sum_{k=i-1}^{j} \left(depth(d_k) + 1\right).q_k$$

$$E[i,j] = \sum_{k=i}^{j} depth(k_k).p_k + \sum_{k=i-1}^{j} depth(d_k).q_k + \sum_{k=i}^{j} p_k + \sum_{k=i-1}^{j} q_k$$

$$E[i,j] = \sum_{k=i}^{j} depth(k_k).p_k + \sum_{k=i-1}^{j} depth(d_k).q_k + w(i,j)$$

Thank you

Please send your feedback or any queries to akyadav1@amity.edu, akyadav@akyadav.in or contact me on +91 9911375598