CS 721: Advanced Algorithms & Analysis

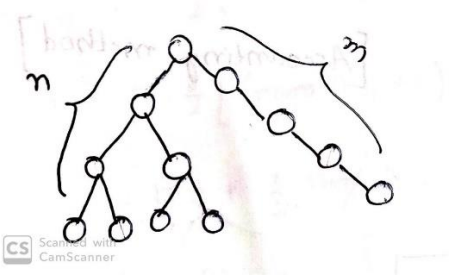
Homework – 2

Chakradhar Reddy Donuri

E949F496

1. (20 points) Suppose T is a binary search tree with m nodes, A is a an array with n elements, and a procedure P employs the Tree insert operation of Page 294 of your textbook repeatedly to insert all elements of A into T, one after another. What is the best case and worst case running time of this operation? Express your answer in Big-Oh or _ notation in terms of m and n and provide brief justification for your answer.

Answer: Best Case: When Tree 'T' is a chain and all the nodes that are attached forms a balanced Binary tree starting at the other child node.



(n/2)logn + (n/4)logn – 1 + (n/8)logn-2 + …..

Therefore the Run time is given as O(nlogn -n)

Worst Case: When the Binary Tree 'T' is a long chain and adding new elements make the chain even longer. Consider the insertion, where it needs to traverse the complete depth of the tree before inserting the i^{th} element from A. This takes cost of (m + i)

Total Cost = $\sum$ i through n (m + i) = $\sum$ i through n (m) + $\sum$ i through n (i)

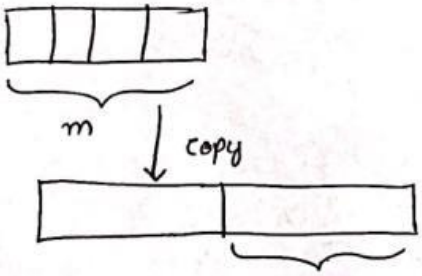
= mn + n(n+1)/2

Therefore the run time is $O(mn + n^2)$

2. (40 points) In this problem you will consider dynamic tables where only insertions are allowed. Initially the size of the table is 1. The cost of insertion is 1 if the table is not full. When an item is inserted into a full table, it first expands the table and then inserts the item into the new table. The expansion is done by allocating a table of size 3 times larger than the old and copying all elements of the old table into the new table.
(a) Calculate the amortized cost using accounting method.
(b) Calculate the amortized cost using potential function method by defining appropriate potential function.

Answer:

2) a) when the table is full, expand its size to 3 times of the earlier size [Accounting method]



Consider the total charge be ϕ_c^{2m}

Total amount for '2m' insertion (3m-m) remaining size of the table is

$$2m \times c - 2m = 3m$$

$$2mc = 5m$$

$$c = 5/2 \;//$$

2) b) We have num(T) elements in the table num(T) - size(T)/3 insertions have each contributed 3/2 to potential in system $\left(\frac{5}{2} - 1 = \frac{3}{2}\right)$ [Potential method]

∴ we can define our potential function as

$$\phi(D_i)_t = \frac{3}{2}\left(num(D_i) - \frac{size(D_i)}{3}\right)$$

case ① :- No expansion

$$\hat{C}_i = C_i + \phi D_i - \phi D_{i-1}$$

∵ no expansion $C_i = 1$

$$size(D_i) = size(D_{i-1}) \rightarrow ①$$

$$Num(D_i) = Num(D_{i-1}) + 1 \rightarrow ②$$

$$C_i = 1 + \frac{3}{2}\left(num(D_i) - \frac{size\ D_i}{3}\right) - \frac{3}{2}\left(num\ D_{i-1} - \frac{size\ D_{i-1}}{3}\right)$$

$$C_i = 1 + \frac{3}{2}\left[(num\ D_{i-1} + 1) - \frac{size\ D_{i-1}}{3}\right] - \frac{3}{2}\left[num\ D_{i-1} - \frac{size\ D_{i-1}}{3}\right]$$

$$= 1 + \frac{3}{2} + \frac{3}{2} num\ D_{i-1} - \frac{3}{2}\frac{size\ D_{i-1}}{3} - \frac{3}{2} num\ D_{i-1} + \frac{3}{2}\frac{size\ D_{i-1}}{3}$$

$$= 5/2 \ //$$

If i^{th} table insertion causes table expansion

$num\ D_i = num\ D_{i-1} + 1$, $size\ D_i = 3\ size\ D_{i-1}$

c_i = cost of inserting i^{th} element is $num(D_i)$

$$\hat{c}_i = c_i + \phi(D_i) - \phi(D_{i-1})$$

$$= num(D_i) + \frac{3}{2}\left(num D_i - \frac{size\ D_i}{3}\right) - \frac{3}{2}\left(num\ D_{i-1} - \frac{size\ D_{i-1}}{3}\right)$$

$$= num(D_i) + \frac{3}{2} num D_i - \frac{1}{2} size\ D_i - \frac{3}{2}\left([num\ D_i - 1] - \frac{size\ D_i}{9}\right)$$

$$= num\ D_i + \frac{3}{2} num D_i - \frac{size\ D_i}{2} - \frac{3}{2} num\ D_i + \frac{3}{2} + \frac{size\ D_i}{6}$$

$$= \frac{3}{2} + num\ D_i - \frac{size\ D_i}{3}$$

But $num\ D_i = \frac{size\ D_i}{3} + 1$

$$\Rightarrow \frac{3}{2} + \frac{size\ D_i}{3} + 1 - \frac{size\ D_i}{3} = \frac{3}{2} + 1$$

$$= 5/2 \ //$$

3. (20 points) A pharmacist has W pills and n empty bottles. Let {p1; p2; : : : ; pn} denote the number of pills that each bottle can hold. Bottle i has associated cost ci and the pharmacist

wants to minimize the total cost of the bottles used to store all the pills. Give a bottom-up dynamic programming solution for this problem.

Answer:

- Let C[i, j] = minimum cost obtained by storing 'j' pills in bottles from 1 to i
- C[n, W] will be the Dynamic programming solution
- Solution for the problem C[i, j] can be written in terms of sub-problems
- We have two cases in here whether Bottle ' i ' is used or not used

Case- 1: Bottle ' i ' is used

- Hence, the optimal is e_1 plus optimal cost of storing $j-p_i$ pills in bottles 1 through (i-1)

Case- 2: Bottle ' i ' is not used

- Hence, the optimal is of storing j pills in bottles 1 through (i-1)

Therefore we can write as C[i, j] = min $\{c_i + c[i-1,j-p_i], c[i-1,j]\}$

If $j-p_i$ is negative then above expression doesn't hold good therefore we need to consider $p_i > j$

Hence the expression is written as C[i, j] = min { c_i , c[i-1,j] }, because if bottle i is used it is not filled.

Now, we have C[i, j] = min { c_i , c[i-1,j] } when $p_i > j$

C[i, j] = min $\{c_i + c[i-1 , j-p_i], c[i-1 , j]\}$ if $p_i <= j$

C[i, j] depends on C[i-1 , j] and C[i-1 , t] where t < j, hence all the sub-problem solutions should be available if we fill the table row-wise.

```
For i = 1 to n
        C[ i, 0 ] = 0
For j = 1 to W
        C[ 0, j ] = 0
For i = 1 to n
        For j = 1 to W
                if p_i <= j
                        with = C_i + e[j-1 , j-p_i],
                else
                        with = e_i
                without = e[i-1 , j]
                e[i , j] = min{with, without}
```

Hence the Run time is given as O(nW).

4. (20 points) Answer the following.
(a) Suppose you insert a node with key value 36 in the red-black tree shown on page 310 of your textbook (Figure 13.1). If the inserted node is colored red, will the resulting tree be a red black tree? What if it is colored black?

Answer:

If Colored Red: No, because it will be a red child of the red node which will violate Red-Black true property.

If Colored Black: No, because it will violate the property that all the paths from the root node will have same number of internal black nodes.

(b) Explain why the longest path from a node x in a red black tree to a descendant leaf has length at most twice that of the shortest path from node x to a descendant leaf.

Answer: In the shortest path there will be no red nodes and it contains all black nodes where as in Longest path every alternate node is black i.e. (Red and Black nodes will be in alternative).

(c) What is the largest possible number of internal nodes in a red-black tree with black height k? What is the smallest possible number?

Answer:
Smallest Possible number: 2^k-1

This will eventually happen only when the tree has only black nodes but not red nodes.

Largest Possible number: 2^{2k-1}

This will eventually happen when Red and Black nodes are in alternate positions in every path.

(d) Describe a red-black tree on n keys that realizes the largest possible ratio of red internal nodes to black internal nodes. What is this ratio? What tree has the smallest possible ratio, and what is the ratio?

Answer: The Largest possible ratio of red internal nodes to black nodes is 2 i.e. when each Black node has two Red node children.

The Smallest possible ratio of Red internal nodes to Black nodes is 0 i.e. when there are no Red nodes.