CS-202

ASSIGNMENT-06

S. SAMARTH REDDY
B19109
GROUP-25

(1) Depth / Height of a B Tree:

No. of elements = n ; Max children a node can have = m

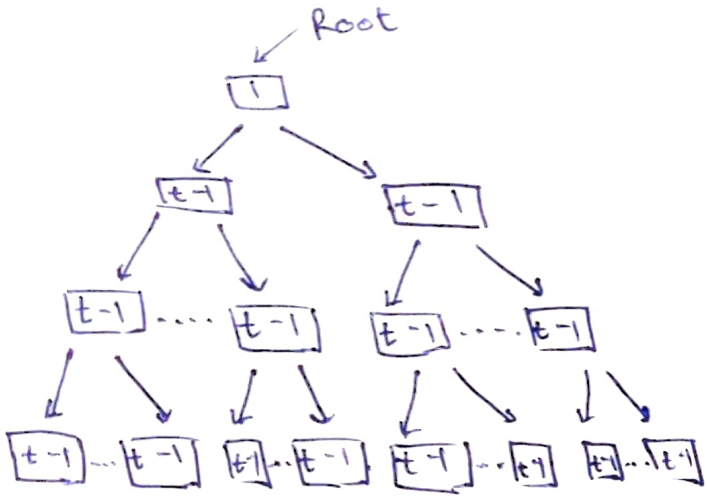
⇒ Worst Case [Maximum Height]

Every non-leaf node (except root) has $t = m/2$ child nodes.

Root node contains 1 element and others $t-1$

Let h be height of the tree. Then

$$n \leq 1 + (t-1)\sum_{i=1}^{h} 2^{i-1}$$



(no. of nodes at depth $h = 2^{h-1}$)

$$\Rightarrow n \leq 1 + 2(t-1)\left(\frac{t^h - 1}{(t-1)}\right)$$

$$\Rightarrow n \leq 2t^h - 1$$

$$\Rightarrow h \geq \log_t \frac{n+1}{2}$$

$$\Rightarrow \boxed{h_{max} = \left\lfloor \log_{\lceil m/2 \rceil} \frac{n+1}{2} \right\rfloor}$$

⇒ Best Case [Minimum Height]

Each node has (m-1) keys (completely filled). Then,

$$\boxed{h_{min} = \lceil \log_m (n+1) \rceil - 1}$$

Since $\left[n = m^{h+1} - 1\right]$ when completely filled

This can be proved by induction.

(2) Let the number of floors in the building be n. We have 2 eggs.

Let the worst case number of drops be x.

⇒ We make our first attempt at x^{th} floor.

If it breaks, we try the remainining $(x-1)$ floors below it one by one, So in worst case we make x drops.

⇒ If it doesn't drop on the x^{th} floor, we jump $(x-1)$ floors (because we used up one trial and we don't want to exceed x). Therefore, if it breaks on the $(x+(x-1))^{th}$ floor, we check the floors $x+1, x+2, \ldots (x+x-1)-1)$. So, in worst case, number of drops is still x.

⇒ Similarly, if the egg does not break on $(x+(x-1)^{th}$ floor, we jump $(x-2)$ floors, so $x+(x-1)+(x-2)$. And then $x+(x-1)+(x-2)+(x-3)$.

⇒ We keep doing this, and in the worst case,

Last floor tried $= x+(x-1)+(x-2)+\ldots+1$.

$$= \frac{x(x+1)}{2}$$

$$\frac{x(x+1)}{2} \geq \text{No. of floors in building} = n$$

$$\Rightarrow x^2+x-2n \geq 0 \Rightarrow x = \left\lceil \frac{-1+\sqrt{8n+1}}{2} \right\rceil$$

Hence, x = min. no. of Sufficient drops $= \Theta(\sqrt{n})$

(3) Problem- Given an array of n real numbers, find the maximum sum by any contiguous subarray

We solve this using Dynamic Programming in bottom-up manner.

Let's say, $(MaxSum)_i$ is the maximum sum by any contiguous subarray ending at index i, then

$$(MaxSum)_i = Max\left(A[i],\ (MaxSum)_{i-1} + A[i]\right)$$

This is a DP solution as we solve the main problem using the overlapping subproblems.

We can make an array for MaxSum[n] and traverse through it and store the values as mentioned above. Then to find the maximum subarray sum. We traverse through it and find the max.

Or, we can also use the algorithm below:-

```
MaximumSum (A)
    max_so_far = 0
    max_current = 0
    Iterate each element from index i = 0 to i = n-1
        max_current = max(max_current + A[i], A[i])
        max_so_far = max(max_so_far, max_current)
    return max_so_far
```

As we go through each element once, $T(n) = \Theta(n)$

(4) Given: $G(V, E)$ → weight directed graph.

- edges leaving Source may have negative weights
- all other edges - non negative weights
- no negative weight cycles.

To Prove: Dijkstra's algorithm finds shortest paths from S correctly in this graph.

Proof: we know that

(i) there are no negative weight cycles in G.

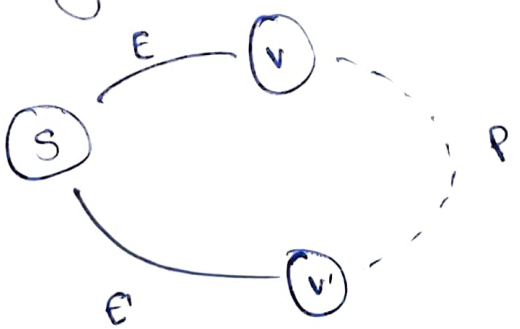
(ii) all negative weights are connected to the Source vertex S.

Therefore, we just need to Prove that:

(i) if Some vertex $v \neq s$ is connected with Some negative weight edge e, the Shortest Path from s to v must Cover the negative weight edge e.

(ii) Apply theorem of correctness of Dijkstra's algorithm and know that Dijkstra's algo is Still Correct.

Proof by contradiction:



[Both E & E' are negative weight edges]

Assume the Shortest Path from v to S is

$S \xrightarrow{E'} v' \xrightarrow{P} v$ instead of $S \xrightarrow{E} v$.

Then we get the equation,

$$E' + P < E$$

$$E + E' + P < 2E < 0$$

This means a negative weight cycle.

This is a contradiction. as we know there are no negative weight cycles in G_f.

∴ Dijkstra's algorithm correctly finds the Shortest Path from s in this graph.

(5) We analyse the worst case Scenario:

Let the Source be 'S and the destination be T.

Let P be the Shortest Path from S to T.

Let x be any vertex on P.

⇒ We claim that the Shortest Path from S to x in the graph is the Path from S to x in P. This is true, because if there was a Shorter Path in the graph from S to x, we would have chosen that Path and then the Path from from x to T in P. But that is a contradiction, as P is the Shortest Path.

$$P(s,t) = P(s,x) + P(x,t)$$

⇒ Hence, we Proved that the Shortest Path from S to x in the graph, is the Path from S to x in P.

⇒ Therefore, in the worst-case Scenario, the Shortest Path from S to Some vertex v, may have to Pass through all the other vertices in the graph. (which means we find the Shortest distance to all Points).

∴ The complexity [in worst case] of finding the Shortest Path to Some vertex is as hard as finding the Shortest Path from S to all other vertices.