CS 570 HW 1

Mounika Mukkamalla

August 2023

1 Time Complexity Order

$O(log(n)) = O(log(log(n^n))), 2^(log(n)), -O(log(n!)) = O(nlog(n^2)), 2^(3n), 3^(2n), n^(nlog(n)), n^(n^2)$

2 Let's prove the given statement by induction.

Base Case: For $k = 1$, We need to prove $k^3 + 5k$ is divisible by 6.
By substituting $k = 1$, $k3 + 5k = 1 + 5 = 6$
6 is divisible by 6.
Hence base case is proved.
Inductive Hypothesis: Assume ∃ r for which given statement is true.

$$\exists r, p \in N \ni r^3 + 5r = 6p$$

Inductive Step: We need to prove the given statement holds true for $r + 1$
By substituting $r+1$ in given expression: $(r + 1)^3 + 5(r + 1) = r^3 + 3r^2 + 3r + 1 + 5r + 5 = (r^3 + 5r) + (3r^2 + 3r^2 + 6)$
$= 6p + 3r(r + 1) + 6 — 1$
Here, consider $r(r + 1$, We know that in a pair of consecutive integers, exact one integer is divisible by 2.
Hence $\exists c \in N \ni r(r + 1) = 2c$
Therefor, eq 1 becomes: $6p + 3(2c) + 6 = 6(p + c + 1)$
Hence proving the given statement is true for $r + 1$.
Inductive Step is proved. Given statement holds true $\forall r \in N$

3 Given equation: $1^3 + 2^3 + ... + n^3 = n^2(n + 1)^2/4$

Base Case: for $n = 1$,
$LHS = 1^3 = 1$
$RHS = 1^2(1 + 1)^2/4 = 1$
LHS = RHS.
Hence given equation is true for base case.

Inductive Hypothesis: Assume $\exists r \in N$ for which given equation is true.
$1^3 + 2^3 + ... + r^3 = r^2(r+1)^2/4$
Inductive Step: We need to prove if the given equation is true for $r+1$.
LHS: By substituting $r+1) : (1^3 + 2^3 + ... + r^3 + (r+1)^3$
$= r^2(r+1)^2/4 + (r+1)^3$ RHS: By substituting $(r+1) : ((r+1)^2(r+2)^2)/4 =$
$((r^2)(r+1)^2 + (4r+4)(r+1)^2)/4$
$= r^2(r+1)^2/4 + (r+1)(r+1)^2$
$= r^2(r+1)^2/4 + (r+1)^3$
LHS = RHS
Inductive Step is true. Hence given equation is true $\forall n \in N$

4 DO IT

5

There are total 5 paths in which Amy can go from Home(H) to SGM(S). Out of which 2 paths are shortest paths with 21 time cost. Below is the list of all 5 paths.
1. $H- > A- > B- > F- > S$
2. $H- > A- > B- > c- > S$
3. $H- > D- > B- > C- > S$
4. $H- > D- > B- > F- > S$
5. $H- > D- > E- > F- > S$

1 and 5 are the shortest paths for Amy.

6 Dooooo

7

In order to prove the given 2 statements, we need a prove:
$numberofnodesinonelevel = sumofallthenodesinthepreviouslevels + 1$
– Statement 0 Assume the given graph contains l number of levels.
Since it is complete binary graph, every node has 2 children, hence number of nodes at a level l_1 is twice the number of nodes in the previous level.
So, number of nodes in each level are: $1, 2, 4, ...2^l_1$
Number of nodes in $l_1 + 1$ level, $N_(l_1 + 1) = 2^(l_1 + 1)$
Sum of all the nodes till $l_1 : S_(l_1) = 1 + 2 + 4 + ... + 2^(l_1) = 2^(l_1 + 1) - 1$
$S_(l_1) = N_(l_1 + 1) - 1$
Hence Statement 0 is true.

St 1:
Left most node is labeled as t.
According to the numbering given in the question, Sum of all the nodes in the

previous levels would be t.
Based on our previous proof, $numberofnodesincurrentlevel = t + 1$
Hence, $lastnodeofthecurrentlevel = leftmostnode+numberofnodesinbetween-1 = t + (t + 1) - 1 = 2t$
Left most child node of node $'t'$ is next node of right most rode in the current level.
Hence Left most child node = right most node +1
Left most child node = 2t+1.

St 2:
Suppose we are at a level l, and consider node t somewhere in the mid of the level.
Let r be the left most node of level l.
Number of nodes from node r to node t excluding t including r = t-r+1
Since graph is complete binary, all the nodes before t from r have 2 children.
Number of children for all the nodes before t = 2(t-r+1)
From St 1, We know that left most child of node r is $2r + 1$
Hence left most child of node t, C_t = left most child of node r, C_r + number of nodes in between $C_t and C_r$ - 1
$C_t = (2r + 1) + 2(t - r + 1) - 1$
Hence, $C_t = 2t + 1$
Proved!

8 Do 8

9

$\exists k \ni 2^k < n$
Let's denote operation cost of $i^t h$ operation as O_i
Sum of all operations in a sequence of n operations, $S_n = O_1 + O_2 + ...O_n$
Given $O_(2^r) = 2^r$, and for all other i, $O_i = 1$
Sum of all (2^j) operations for integer $0 <= j <= k$,$S_j = O_(2^0) + O_(2^1) + ...O_(2^k) = 1 + 2 + 4 + ...2^k = 2^(k + 1)$
Sum of all the remaining operations, $S_r = n - k - 1$
$S_n = S_j + S_r = 2^(k + 1) + n - k - 1$
$= 2(2^k) + n - k - 1 < 2(n) + n - k - 1 < 2(n) + n = 3n$
Hence, $S_n < 3n$
Amortised cost = Sum of all operations cost / number of all operations.
=S_n/n ¡ 3 = O(1).
Hence Amortised cost per operation is constant, O(1).

10

Operation cost of insert operation, O_i: constant = O(1)
$O_i < c_1$
After n insert operations,
Operation cost of lookup operation, O_l: linear time = O(n)
$O_l < c_2 n$ Total operation cost = (Sum of all insert operations cost + lookup operation cost)
= $(n(O_i) + O_l)$ ¡ $nc_1 + c_2 n = (c_1 + c_2)n$
Total Operation cost = O(n) ¡ cn
Amortised cost = Total Operation cost/ Number of operations
= $cn/(n+1)$ = O(1)
Amortised cost per operation = O(1)