11-09 Doubt Session

Sunday, 11 September 2022 10:12 AM

$T(n) = n * T(n-1)$] — (1)

$T(n-1) = (n-1) * T(n-2)$ — (2)

2 into 1

$T(n) = n * (n-1) * T(n-2)$

Putting n-2 in 1 $T(n-2) = (n-2) * T(n-3)$ — (3)

$T(n) = n * (n-1) * (n-2) * T(n-3)$ —

$= n\,(n-1)(n-2)(n-3) \cdots T(n-(n-1))\ T(1)$

$= n(n-1)(n-2)(n-3) \cdots 1 \quad T(n-n+1)$

$= n(n-1)(n-2)(n-3) \cdots 3 \times 2 \times 1$

$= n * n\left(1-\frac{1}{n}\right) * n\left(1-\frac{2}{n}\right) * n\left(1-\frac{3}{n}\right) \times \cdots \times n\left(\frac{2}{n}\right) \times n\left(\frac{1}{n}\right)$

$= n \times n \times n \times n \times n \times n \cdots$ —

$= n^n$

$2 \downarrow 2+3$

3

2

$2 \overset{+3}{\underset{2}{2}}$

1. O notation:-

- This notation is used to express upper bound of an algorithm's running time. ✓ tight
- It represents the worst case of algo's time complexity. i.e. the longest amount of time an algo possibly can take.

$f(n) = O(g(n))$; if there exist a value +ve int $n \geq n_0$

$f(n) \leq c\,g(n) \quad \forall\ n \geq n_0,\ c > 0,\ n \geq 1$

$f(n) = 2n^2 + 5n + 1$ $g(n)$, c, n_0

$g(n) = n^2$

$c = 2$

$f(n) \leq c\,g(n)$

$2n^2 + 5n + 1 \leq 3n^2$ $\quad n=1$

$2+5+1 \leq 2 \times 1$

$8 \leq 2$ ××

$n=1$

$8 \leq 3 \times 1$ ≥ 8

$2n^2 + 5n + 1 \leq 8n^2$

$c = 8$

$g(n) = n^2$

$n=1$

$2+5+1 \leq 8$

$8 \leq 8$ } true

$2n^2 + 5n + 1 \leq 4n^2$

$n=1 \quad 8 < 4$ false

$2n^2 + 5n + 1 \leq 3n^2$

$n=1 \quad 2+5+1 \leq 3$ false

$n=6$? ?

$2n^2+5n+1 \le 3n^2$

$\forall n \ge 6$

$2(6)^2+5(6)+1 \le 3(6)^3$

$72+30+1 \le 108$

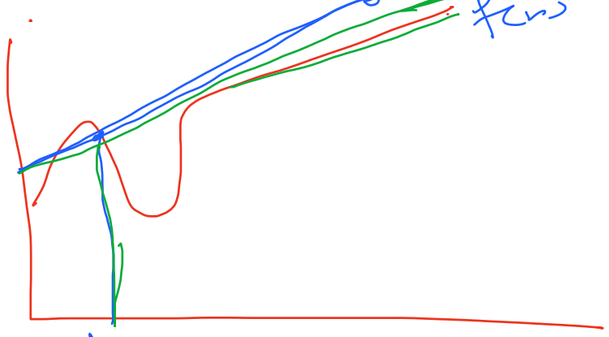
$103 \le 108$ ✓

$n=5$ $\quad 2(25)+25+1 \le 6(25)$

$50+25+1 \le 150$ ✓ $\quad C=3$

$n=4$ $\quad 2(16)+20+1 \le 6(16)$

$32+20+1 \le 96$ ✓



$2n^2+5n+1 \le 8n^2$

$n=1$

$2+5+1 \le 8$

$8 \le 8$

$C=8$

$g(n)=n^2$

$n \ge 1$

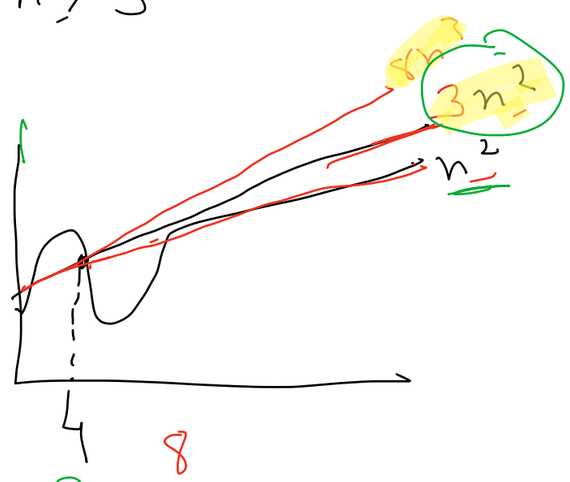
$n=1$ $\quad 8 \le 4$

$2n^2+5n+1 \le 3n^3$ ✓

$C=3$

$g(n)=n^2$

$n \ge 5$



(a) $C = g(n)$

(b) $C = g(n)$

(c) $C = g(n)$

(d) $C = g(n)$

Q $f(n) = 2n^2+5n+1$, $g(n)=n^3$ $\quad C=$

$2n^2+5n+1 \le C\,n^3$

$2n^2+5n+1 \le 8n^3$

$C=8$

$n=1$

$2+5+1 \le 8$ ✓

$2n^2+5n+1 \le 4n^3$

$C=4$

$n=1$

$2+5+1 \le 4$ false

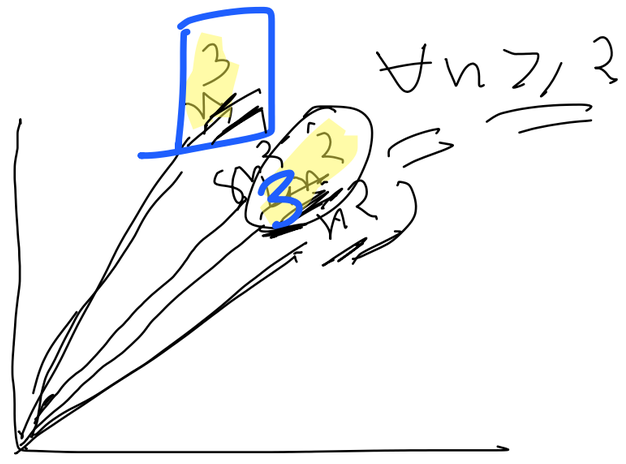
$4n^3 \le 8n^3$ ✓

$4n^3$ ✓

$n=2$

$2\times4+5\times2+1 \le 4\times8$

$8+10+1 \le 32$ true



$\forall n \geq 2$

$c=6, n=1$

$6n^3 \geq 4n^3$

(1) Loop:-

```
for(i=1; i<=n; i++)
{
    x = y+2 // c
}
```

$O(cn)$

$O(n)$

(2) nested loop:

```
for(i=1; i<=n; i++)
{
    for(j=1; j<=n; j++) { // n times
        x = y+2 // constant time
    }
}
```

$O(cn \times n)$ $O(cn^2)$ $O(n^2)$

(3) Sequential

```
a = a+b; // c1
for(i=1; i<=n; i++) //n
{
    x = y+z; // c2
}
for(j=1; j<=n; j++) //n
{
    c = d+e; // c3
}
```

$c_1 + nc_2 + nc_3$ $n+n$ $O(n)$

(4) if-else

```
if (condition)
{
    ---  //O(n)
}
else
{
    ---  O(n^2)
}
```

$O(n)$ $O(n^2)$ $O(n^2)$

Q. A child is running up staircase with n steps and can hop either 1 step, 2 steps or 3 steps at a time. Implement a method to count

how many possible ways a child can run up the stairs.

n=1 → 1 { Possible no. of :- 1 (step of 1) case

n=2 → 2 { 1+1 (2 steps of 1 each)
2 (1 step of 2)

n=3 → 4 { 1+1+1 (3 steps of 1 each)
2+1 (first => 2, second (1)
1+2
3 (one step of 3) → 1 } } ④

n=4 → 7 { 1+1+1+1, 1+1+2, 1+2+1, 2+1+1,
3+1, 1+3, 2+2

function findStep(n)

n = 3 S

{ 101 010 000
100 001 }

function

JS
Adv. JS
Arrays
Alg. ana