17-1-2020

Homework - 1

R.Abhinav.
CB.EN.U4CSE19453

1) Show that $6n^2+20n$ is $O(n^3)$

Sol:- let $f(n) = 6n^2+20n$

$g(n) = n^3$

as $O(n^3)$ is given

$f(n) \leq C * g(n)$

$\frac{f(n)}{g(n)} \leq C$ $\frac{g(n)}{g(n)} = C$

$f(1) = 6+20$

$f(1) = 26$

let $c = 26$

$6n^2+20n \leq 6n^3$; $n \geq 1$

by dividing with n^3;

$\frac{6}{n} + \frac{20}{n^2} \leq 6$; $n \geq 1$

i and $n = 1$

$6+20 \leq 6$

$26 \leq 6$

as it fails th $n \geq 1$ condition

$6n^2+20n$ is not $O(n^3)$

it is $O(n^2)$

hence $6n^2+20n$ is $O(n^2)$

$f(n) \in O(g(n))$

$\boxed{6n^2+20n \in O(n^2)}$

2.) (a) boolean subset (int [ ] sub, int [ ] super) {

```
    int  m = sub.length;
    for (int i=0; i<m, i++)
        if (!member (sub[i],super) return false;
    return true;
}
```

Sol: $1+1+m+1+m+m+m+(1)$

$= (4+4m)$

(b) boolean member(int x, int [ ], a) {

```
int n = a.length;
for (int i=0; i<n ; i++) {
   if (x == a[i]) return true;
}
return false;
}
```

Sol: $3+4n$

Complexity $= a + mb$

$= (4+4m) + m(3+4n)$

$= 4+4m+3m+4mn$

$f(n,m) = 4+7m+4mn$

in big O notation $O(n*m)$

3.) Prove that $f(x) = 4x^2 - 5x + 3 = O(x^2)$

Sol:
$$|f(x)| = |4x^2 - 5x + 3|$$
$$\leq |4x^2| + |-5x| + |3|$$
$$\leq 4x^2 + 5x + 3 \quad ; \text{ for all } x > 0$$
$$\leq 4x^2 + 5x^2 + 3x^2 \quad ; \text{ for all } x > 1$$
$$\leq 12x^2 \quad ; \text{ for all } x > 1$$

we conclude that $f(x)$ is $O(x^2)$

observe that $C = 12$ and $K = 1$ from the def^n of Big O.

4.) ⓐ $n^2 + 3n$

The highest degree of n is 2.

So big O notation is $O(n^2)$

Proof:
$$f(n) = n^2 + 3n$$
$$g(n) = n^2 \; ; \; n \geq 1$$

$$f(n) \leq c\,g(n)$$
$$f(1) = 4$$
$$c \geq 4$$
$$n^2 + 3n \leq 4n^2 \quad ; n \geq 1$$

divide with n^2

$$1 + \frac{3}{n} \leq 4 \quad ; n \geq 1$$

$$1 \leq 4 - \frac{3}{n}$$

$n = 1$; $1 \leq 1$

$n = 2$; $1 \leq 4 - 1.5$

$1 \leq 2.5$

$n \geq 1$;

always satisfied for higher values

So complexity is $O(n^2)$.

4.)

ⓑ $3n^2+112n$

Complexity is $O(n^2)$

Proof:

$f(n) = 3n^2+112n$

$g(n) = O(n^2)$

$f(n) \le c \cdot g(n)$

$f(1) = 3+112$

$f(1) = 115$

$\boxed{c \ge 115}$

5.)

b.)
```
{
  int z=a+b+c;
  return (z);
}
```

$T = \underbrace{O(1)}_{c_1} + \underbrace{O(1)}_{c_2}$

adding $c_1, c_2 \rightarrow c_3$

$T = c_3$

implies $O(1) \rightarrow$ Constant time.

5) (a) int sum (int a[], int n)

```
{
① int x = 0;
② for (int i=0; i<n; i++)
  {
③   x = x + a[i];
  }
④ return(x);
```

Sol: ① initialization takes same amount of time every time $O(1)$

③ its takes same time to execute throughout the loop

so → $O(1)$

Now, since loop runs for n times

time complexity eqⁿ(1) $= O(1) + n[O(1)]$

④ return will also take same time $O(1)$ to get value from variable.

$$T = O(1) + n[\underbrace{O(1)}_{C_1} + \underbrace{O(1)}_{C_2}]$$

C_1 ↓ let us take as constant

adding;

$C_1, C_2 \Rightarrow C_3$

$T = C_3 + n(C_4)$

fasting growing cup.

removing coeff. $n(C_4)$

then

big O of eqⁿ : $n \rightarrow O(n)$ "$\boxed{\text{Linear}}$"

6.) Outer for loop:

$(1) + \left(\frac{n}{2}+2\right) + 2\left(\frac{n}{2}+1\right)$

$\frac{n}{2}+2+n+2$

$5+n+\frac{n}{2}$

inner for loop:

$(1)\ (n+1)\ (n)\ (2n)$

$= 1+n+1+n+2n$

$= 2+4n$

$= 1+5+n+\frac{n}{2}+\frac{n}{2}(2+4n)+1$

$= 1+5+n+\frac{n}{2}+\frac{2n}{2}+2n^2+1$

$= 7+n+\frac{n}{2}+n+2n^2$

$= 7+2n+\frac{n}{2}+2n^2$

$f(n) = 7+\frac{5n}{2}+2n^2$

Complexity is $O(n^2)$

$3n^2+112n \leq 115n^2$; $n \geq 1$

divide by n^2

$3+\frac{112}{n} \leq 115$; $n \geq 1$

$3 \leq 115-\frac{112}{n}$; $n \geq 1$

if $n=1$

$3 \leq 3$

$n=2$

$3 \leq 115-56$

$3 \leq 59$; $n \geq 1$

$n \geq 1$;

condition always satisfied

hence complexity is $O(n^2)$

Scanned by CamScanner

7.) a) $f_1(n) = 2^n$; $O(2^n)$

b) $f_2(n) = n^{3/2}$; $O(n^{3/2})$

c) $f_3(n) = n\log n$; $O(n\log n)$

d) $f_4(n) = n^{\log n}$; $O(n^{\log n})$

- $n = 1$:

$f_1(1) = 2$

$f_2(1) = 1$

$f_3(1) = 0$

$f_4(1) = 1^0 = 1$

- $n = 10$:

$f_1(10) = 2^{10} = 1024$

$f_2(10) = 10^{3/2} = 31.622$

$f_3(10) = 10\log(10) = 10$

$f_4(10) = 10^{\log 10} = 10$

- $n = 100$:

$f_1(100) = 2^{100} = 1.26 \times 10^{30}$

$f_2(100) = 100^{3/2} = 1000$

$f_3(100) = 100\log(100) = 200$

$f_4(100) = 100^2 = 10000$

$f(3) < f(2) \leq f(4) < f_1$

So:

$O(n\log n) < O(n^{3/2}) < O(n^{\log n}) < O(2^n)$

8.) for (int i=1; i<=m; i+=c)
{
for (int i=1; i<=n; i+=c)
}

Sol:- for (int i=1; i<=m; i+=c)
{
O(1) (m)
}

for (int i=1; i<=n; i+=c)
{
O(1) (n)
}

(1+m+1+2m+m) + (1+n+1+2n+n)

(2+4m) + (2+4n)

f(m) + f(n) = 2+4m+2+4n

= 4m+4n+4

in O notation;

Complexity is O(m)+O(n)

9.) To prove : $O(1) + O(1) = O(1)$

These are constants

$c_1 + c_2 = O(1) + O(1)$

$c_1 + c_2 = c_3$

c_3 is also a constant.

$c_3 = O(1)$.

hence proved

$O(1) + O(1) = O(1)$

10.) Time Complexity :-

- The algorithm that has least run time
- Time complexity is asymptotic notation for n inputs
  eg : $O(n)$
  $n = 1, 2, 3, \ldots -$

Total Execution time :-

- How long the program runs
- Total time for a program to execute.
- eg :- $f(n) = n^2 + 3n$.

Scanned by CamScanner