Ahmad Abdullah
i22-1609.

Design Analysis & Algorithms

Assignment #1

Question #1: Calculate time complexity function and Big Oh.

→ A) int Function (Array A <int>) ...

$F(n) = 1+1+1+1+n+n+n+n+n+n+n+n+n+n+1+1$

$\Rightarrow F(n) = 6 + 10n$

$\Rightarrow O() = O(n)$

→ B for i = 1 to n do ...

$F(n) = (n+1)(2n+1+2n+1)$
$= (n+1)(4n+2) \Rightarrow 4n^2+6n+2$

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

→ C for i = 1 to n*n do ...

$F(n) = (n+1)((n+1)+n)$
$= (n+1)(2n+1) \Rightarrow 2n^2+n+1$

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

→ D) for i = 0 to m do ——

$F(n) = (m+1) + m + m\log m + m\log m + m\log m$

$= 2m + 1 + 3m\log m$

$F(n) = O(m\log m)$

—— o

→ e) Sum = 0 ——

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

$= n + 1 + n^2 + n + \frac{n^3 + n^2 + n^2 + n}{2}$

$= \frac{2n + 2 + 2n^2 + n + n^3 + n^2 + n^2 + n}{2}$

$= \frac{n^3 + 4n^2 + 4n + 2}{2}$

$F(n) = O(n^3)$

o —— o

→ 7) proc Matrix Mult (A, B) ——

$F(n) = n^2 + n + n(n+1) + n(n+1) + n + n(n+1) + n^2(n+1)$

$= n^2 + 4n + n^2 + n^2 + n^2 + n^2 + n^3 + n^2$

$= n^3 + 6n^2 + n$

$F(n) = O(n^3)$

o —— o

Question #3 Biryani Langar
But this time, you are the one distributing

```
int total_plates.

function ( Array People-ID <int>)
    ~~for i=0 to n do~~
    while ( PeopleIn_Line) do
        People_ID[i] ++;
        total_plates ++;
    end while;

    for i=0 to n do
        if People_ID[i] > total_plates/2
            print( People_ID [i] ;
            return;

    end for
```

Question # 2

a) $100n + 5 \neq \Omega(n^2)$

We know

$Fn \geq c\,g(n)$ in this case $g(n) = n^2$

$100n + 5 \geq cn^2$

$\frac{100n}{c} + \frac{5}{c} \geq n^2$

which is not true as value of n increases n^2 will grow exponentially, Hence

$F(n) = 100n + 5 \neq \Omega(n^2)$

b) Theta bound for $f(n) = \frac{n^2}{2} - \frac{n}{2}$

We know

$c_1 g(n) \leq f(n) \leq c_2 g(n)$

Lower bound $\leftarrow$ $\frac{n^2}{2} - \frac{n}{2} \leq F(n) \leq \frac{n^2}{2}$ $\rightarrow$ upper band

For upper bound

$\frac{n^2}{2} - \frac{n}{2} \leq \frac{n^2}{2}$

$1 \leq \frac{n}{2}$ which is true for every value if $n \geq 2$ So $F(n) = O(n^2)$

For lower bound

$\frac{n^2}{2} - \frac{n^2}{2} \leq \frac{n^2}{2} - \frac{n}{2}$

$1 \leq 1$ So ~~$f(n) = O(n^2)$~~

So $F(n) = \Omega(n^2)$

Since $\Omega(n^2) = F(n) = O(n^2)$

$n^2 \le F(n) \le n^2$

Theeta bound will be n^2.

c) Prove that $6n^3 \neq \Theta(n^2)$

We can assume that both upper and lower bound are n^2 which will automatically mean $\Theta(n^2)$ is.

So,

$n^2 \le 6n^3 \le n^3$

For lower bound

$n^2 \le 6n^3$ which is true

For upper bound

$6n^3 \le n^2$ which is false and does not hold

so

$6n^3 \neq \Theta(n^2)$

d) Prove $f(n) = n \neq \Theta(\log n)$

Assuming that upper and bound is $\log n$ for the given function

So,

$$\log n \leq n \leq \log n$$

this holds for lower bound but fails to be true for Upper bound

So

$f(n) = n \neq \Theta(\log n)$.

Tamatar Bare Mazedar.