q-3-2

# Assignment 2

## Pseudo Code

```
for (i = 1; i <= n; i = i * 2)
    for (j = 1; j <= i; j++)
        print ("Hello")
```

## Dry Code (for n = 10)

| i | i <= n | j-loop | print | i = i * 2 |
|---|---|---|---|---|
| 1 | ✓ | 1 to 1 | 1 time | 1 × 2 = 2 |
| 2 | ✓ | 1 to 2 | 2 times | 2 × 2 = 4 |
| 4 | ✓ | 1 to 4 | 4 times | 4 × 2 = 8 |
| 8 | ✓ | 1 to 8 | 8 times | 8 × 2 = 16 |
| 16 | X | | | |

We see, the no. of iterations in i:

$1 + 2 + 4 + 8 + \dots + 2^k \leq n$ , (k is the iterations in i)

This is equal to the ~~no. of~~ total no. of iterations in j loop ~~for each~~

Thus, total no. of times of "Hello" is printed in worst case is:

$1 + 2 + 4 + 8 + \dots + 2^k = n$

or $1 + 2 + 2^2 + 2^3 + \dots 2^k = n$

This is a GP (Geometric Progression) with $a = 1$, $r = 2$

Sum of GP, $S_k = \frac{a(r^k - 1)}{(r-1)} = n$

$= \frac{1(2^k - 1)}{2 - 1} = n$

$\Rightarrow 2^k = n + 1$

$k \log 2 = \log (n+1)$

$\Rightarrow k = \log_2 (n+1)$

Thus, time complexity,

$$\boxed{T(n) = O(\log_2 (n+1))}$$