

CS 310 Algorithms

Assignment -1

Name: Abdul Manan

Roll No. 2020-10-0198

Q4-i What is the asymptotic time complexity of the following program fragment.
Show your working.

```
for (i=1; i<=n; i*=2) {  
    j = i;  
}
```

Sol: The time complexity of this snippet is $O(\log n)$.

The value of i is increasing as follow:

1, 2, 4, 8, 16, 32, 64

So, the above loop will run $\log(n)$ times and performs a constant time operation each time.

* * * * *

Q4-ii What is the asymptotic time complexity of the following program fragment.
Give both the upper bound and lower bound for this fragment. Show your working.

Sol:

The outer loop is running ----- n times

The inner loop is running ----- $n (n-1) (n-2) (n-3) \dots 1 = n (n-1) / 2$

The inner function has time complexity of $O(n)$, as it depends on values of i and j .

So, the overall time complexity will be:

Upper Bound: $O(n * n * n) = O(n^3)$

Lower Bound: $\Omega(n^3)$

* * * * *

Q4-iib What is being stored in the 2-D W array in Q4-ia?

Sol:

In the 2-D W array, Upper Triangular Matrix is stored.

The code is converting 1-D array into 2-D Array that is some sort of Prefix Sum.

* * * * *

Q4-iic The program fragment in Q4-ia is not very efficient. Rewrite it to improve its time complexity.

Sol:

Here is the code that does same as the given code snippet but its time complexity is in $O(n^2)$.

```
int x, y, n;
// ...
/* P is a 1-D array of size n integers and
   W is a 2-D array of size n x n integers */

int temp = 0;
for (x=0; x<n; x++) {
    temp = P[x];
    for (y=x+1; y<n; y++) {
        W[x][y] = P[y] + temp;
        temp = W[x][y];
    }
}
// ...
```

* * * * *