Name - Prabhudeep Singh

Tutorial 2

Q1 Initially, $i = 0$, $j = 1$.

	Value of i	Value of j	Sum
1st Time	$i = 1$	$j = 2$	1
2nd time	$i = 3$	$j = 3$	1
3rd Time	$i = 6$	$j = 4$	1
4th time	$i = 10$	$j = 5$	1
5th Time	$i = 15$	$j = 6$	1
6th Time	$i = 21$	$j = 7$	1
...			
Kth Time	$i = n$	$j = K$	1

$$i = \frac{K.(K+1)}{2} = n$$

$$K^2 + K = 2n$$

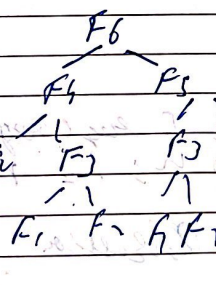
$$K^2 < n$$

$$K < \sqrt{n}$$

Time complexity → O(sqrt(n))

Q2

```
int fib (n)
{
    if (n == 0 || n == 1) return 1;
    else return fib(n-1) + fib(n-2);
}
```



Space complexity:

The space is proportional to the max depth of recursion tree

$T = O(2^n)$

Space complexity $= O(N)$

$T(n) = T(n-1) + T(n-2) + c$

$T(n-1) \approx T(n-2) = 2T(n-2)$

$T(n-2) = 2 * c\ 2T(n-2) + c = 2^2 T(n-2) + c$

$T(n-4) = 2 * (4T(n-2) + 3c) + c$

$= 8T(n-3) + 7c$

let $n - k = 0 \Rightarrow n = k$

$2^k T(0) + [2^n - 1]\, c$

$2^n 1 + 2^n c - c$

$2^n (1 + c) - c$

$\approx 2^n$

Q3 merge sort $= n \log n$

1) nested loop three loops $= O(n^3)$

```
for (int i=0; i<n; i++)
{
    for (int j=0; j<n; j++)
    {
        for (int k=0; k<n; k++)
        {
            // some O(1) expression
        }
    }
}
```

2) For Time complexity [eg (log n)]

```
for (i=2; i<n; i = pow(i,2))
{
    //some O(1) expression
}
```

3) For time complexity $n \log n$ we can use the following function:

```
int fun(int n)
{
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= n; j++)
        {
            // some O(1) expression
        }
    }
}
```

Soln 4 $T(n/2) \geq T(n/4)$

$T(n) = 2T(n/2) + cn^2$

using master's method

$T(n) = aT(\frac{n}{b}) + f(n)$

$c = \log_b^a = 1$

$f(n) = n^c$

$T(n) = \theta f(n)$

$\theta(n^2)$

Soln 5 for $i = 1 \rightarrow j = 1, 2, 3, 4 \dots n$ (Runs for n)

for $i = 2 \rightarrow j = 1, 3, 5 \dots n$ (Runs for $n/2$)

for $i = 3 \rightarrow j = 1, 4, 7 \dots n$ (Runs for $n/3$ times)

$T(n) = n + n/2 + n/3 + n/4 + \dots$

$= n\left(\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots\right)$

$= n \int_1^n \frac{1}{x} = n \int_1^n \frac{dx}{x}$

$= [n \log n]_1^n$

$T(n) = \theta(n \log n)$

Q6

For first iteration, $i = 2$

2nd iteration, $i = 2^K$

3rd iteration, $i = (2^K)^K$

.

.

.

k^{th} iteration, $i = (2^K)^i$

Loop exits when $2^{K^i} = n$

Applying log:

$$\log n = \log 2^{K^i}$$

$$K^i = \log n$$

applying log again

$$i \log K = \log \log n$$

$$i = \log(\log n)$$

$$T(n) = \Theta(\log(\log n))$$

Q7

(a) $100 < \log \log n < \log n < \log^2 n < \sqrt{n} < n$

$< n \log n < n \log n < n^2 < 2^n < 4^n < n! < 2^{2^n}$

(b) $1 < \sqrt{\log n} < \log n = \log n = \log n$

$< 2n = 4n = n < \log n < n \log n < n^2$

$< 2(2^n) < n!$

(c) $2^{\log_2 n} = \log_2 n < 5n < \log n$

$= n \log_2 n = n \log_2 n < n^2 < n^3 < 2^{2^n} < n!$