Tanmay Bhatt
CST SPL-1
24

Sunil
DATE
PAGE
"Rough 'n' Fair"

Tutorial-2

1)
```
void fun(int n)
{
  int j=1, i=0;
  while (i<n)
  { i=i+j;
    j++;
  }
}
```

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

	Value at i	Value at j	Run
1st	i = 1	j = 2	1
2nd	i = 3	j = 3	1
3rd	i = 6	j = 4	1
4th	i = 10	j = 5	1
5th	i = 15	j = 6	1
6th	i = 21	j = 7	1
kth	i = n	j = k	1

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

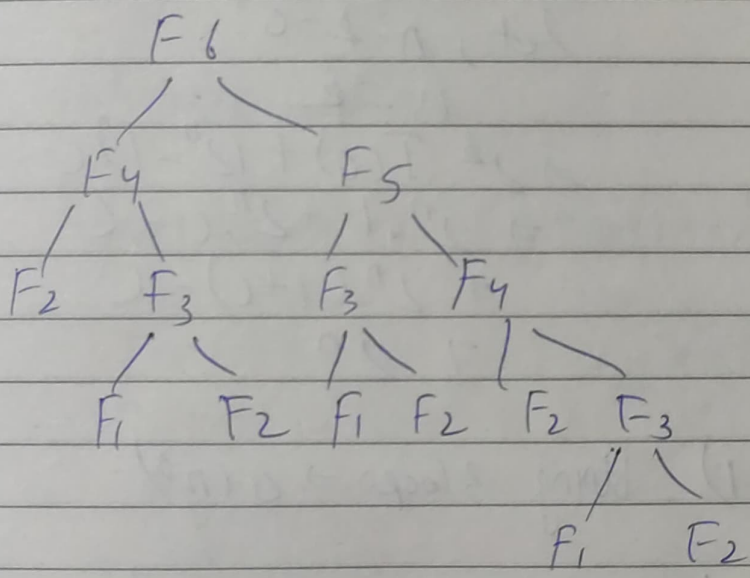
$$k^2 + k = 2n$$

$$k^2 < n$$

$$k < \sqrt{n}$$

Time complexity $\rightarrow O(\sqrt{n})$

2)
```
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 max depth of recursion tree.

$T = O(2^n)$

Space complexity = $O(N)$

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

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

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

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

DATE
PAGE

$T(n-4) = 2 * (4T(n-2) + 3c) + c$
$= 8T(n-3) + 7c$
$= 2^k T(n-k) + (2^k - 1)c$

Let, $n - k = 0$
$n = k$
$2^k . T(0) + (2^n - 1)c$
Let, $n - k = 0$
$n = k$
$\Rightarrow 2^k T(0) + (2^n - 1)c$
$\Rightarrow 2^n . 1 + 2^n . c - c$
$\Rightarrow 2^n (1 + c) - c$
$\Rightarrow 2^n$

3) 1) Using 3 loops $\rightarrow O(n^3)$

```
for (int i=0; i<n; i++)
{
  for (int j=0; j<n; j++)
  {
    for (int k=0; k<n; k++)
    {
    }
  }
}
```

2) Merge Sort : n log n

Sunil
DATE
PAGE
"Rough 'n' Fair"

3) For log (logn) time complexity.

```
for(int i=2; i<n; i=pow(i,c)
  {

  }
```

4) For time complexity : nlogn

```
int fun(int n)
{
  for(int i=1; i<=n; i++)
  {
  for(int j=1; j<=n; j++)
    {

    }
  }
}
```

4) $T(n) = T(n/4) + T(n/2) + cn^2$

$T(n/2) \simeq T(n/4)$

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

Using Master's Method

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

$C = \log 2^2 = 1$

$f(n) = n^2$

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

$\Rightarrow \Theta(n^2)$

5)

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

for $i=1 \rightarrow j = 1, 2, 3, 4, \ldots n$

for $i=2 ; j = 1, 3, 5, \ldots$

for $i=3 \rightarrow j = 1, 4, 7, \ldots n$

$$T(n) = n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \ldots$$

$$\Rightarrow n\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots\right)$$

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

$$\Rightarrow n[\log x]_1^n$$

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

6)
```
for(int i=2; i<n; i<pow(i,k))
{
// Some O(1) expression
}
```

for first iteration $i = 2$
2nd iteration $i = 2^k$
3rd iteration $i = (2^k)^k$
:
nth iteration $i = (2^k)^i$
loop ends at $2^{k^i} = n$

Apply log:
$\log n = \log_2 k^i$

$k^i = \log n$

Applying log again
$i \log k = \log n$
$i = \log(\log n)$

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

8) (a) $n, n!, \log n, \log\log n, \sqrt{n}, \log^2 n$, $n \log n, \log^2 n, 2^n, 2^{2n}, 4^n, n^2, 100$

$\Rightarrow 100 < \log\log n < \log n < \log^2 n < \sqrt{n} < n < \log n! < n \log n < n^2 < 2^n < 4^n < n! < 2^{2n}$

(b) $2(2^n), 4n, 2n, 1, \log n, \log\log n, \sqrt{\log n}, \log 2n, 2\log n, n, \log n!, n^2, n \log n$

$\Rightarrow 1 < \sqrt{\log n} < \log 2n = 2\log n = \log n! < 2n = 4n = n < \log n! < n \log n < n^2 < 2(2^n) < n!$

(c) $(8)^{2n}, \log n, n \log n, n \log_2 n, \log n!, n!, \log_e n^2, 96, 8n^2, 7n^3, 5n$

$\Rightarrow 96 < \log 2n = \log_e n < 5n < n \log_2 n = n \log_e n < 8n^2 < 7n^3 < 8^{2n} < n!$