Design & Analysis of Algorithm

Tutorial - 2

Name ⇒ MOHAK KALA ; Section - D ; Class R. No. ⇒ 29, Univ. R. No. ⇒ 2014727

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

j = 1, 2, 3, 4, 5, 6

i = 0, 1, 3, 7, 12,

TC = O(n)

②
```
void iteration (int n)
{ int f = 1;
  for (i = 2, i<=n; i++)
     f *= i
   cout << f;
}
```

TC = O(n)

SC = O(1)

```
int recursing (int n)
{ if (n<2)
    return 1;
  return n * recursing (n-1)
}
```

TC = O(n)

SC = O(n)

SC is O(n) as stack is created in the memory while recurrence fn. was called till reaching the base condition

③ n log n ⇒
```
for (i=0; i<n ; i++)
    for (j=0; j*j<n ; j++)
        sum = sum + j ;
```

n^3 ⇒
```
for (i=0; i<n ; i++)
    for (j=0; j<n ; j++)
        for (k=0; k<n; k++)
            sum = sum + k;
```

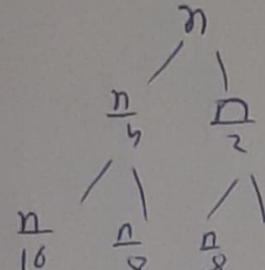
②

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

while $(i > 0)$

$i = \sqrt{i}$;

④ $T(n) = T(n/4) + T(n/2) + cn^2$



$-n^2$

$-\frac{n^2}{16} + \frac{n^2}{4} = \frac{5n^2}{16}$

$-\frac{n^2}{256} + \frac{n^2}{64} + \frac{n^2}{64} + \frac{n^2}{16} = \frac{25}{256} n^2$

$T(n) = c\left(n^2 + \frac{5n^2}{16} + \frac{25}{256} n^2\right)$

$\alpha = \frac{5}{16} \Rightarrow 5\alpha = \frac{1}{1-\alpha}$

$T(n) = cn^2\left(1 + \frac{5}{16} + \frac{25}{256} + \cdots\right)$

$= cn^2\left(\frac{1}{1-\frac{5}{16}}\right) = cn^2 \frac{16}{11}$

$TC = O(n^2)$

⑤ int fun (int n)

for (int i = 1; i < n; i++) $\quad - O(n)$

for (j = 1; j < n; j += i $\quad - \log(n)$

// $O(1)$ $\quad - O(1)$

$TC = O(n \log n)$

⑥ $i = 2, 2^k, 2^{k^2}, 2^{k^3}, 2^{k^4}, 2^{k^5} \cdots 2^{k^x}$

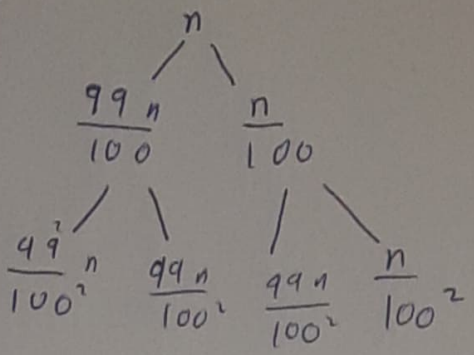
$n = 2^{k^x}$

$\log n = k^x \log 2$

$\frac{\log \log n}{\log 2} = x \log k \Rightarrow x = \frac{\log(\log n)}{\log 2 \times \log k}$

$\boxed{TC = O(\log(\log n))}$

(7)



(3)

Taking Longest branch $= \frac{99n}{100}$

$$TC = \log_{\frac{100}{99}} n = \log n$$

$$n = \left(\frac{99}{100}\right)^k \Rightarrow k = \log_{\left(\frac{100}{99}\right)} n$$

$$T(n) = \frac{n\left(\log_{\frac{100}{99}}\right)^n}{100}$$

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

(8) (a) $1 < \log\log n < \log n < \sqrt{\log n}\ (\text{root}(n)) < n < n\log n < n^2 < 2^n < 2^{2^n} < 4^n < n!$

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

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