Ishwar Singh Bhandari
19011603
A
B. Tech (C.S.E)

Tutorial Sheet - 1

① $O(N+M)$ time
$O(1)$ Space

② $T(n) = O(n)$, Space $O(1)$

③ $T(n) = O(\log_2 n)$, Space $O(1)$

④ int sum = 0, i;
for (i = 0; i * i < n; i++)
{
sum += i;
}

$= n + (n-1) + (n-4) + (n-9) + \dots + (n-k)$

$= n + (n * k) - (1^2 + 2^2 + 3^2 + \dots + k^2)$

$= \sqrt{n}$

$i^2 < n$

$i < \sqrt{n}$

$T(n) = O(\sqrt{n})$, space $O(1)$

⑤ int j = 1, i = 0;
while (i <= n)
i = i + j;
j++; }

$0 <= n \quad 1$
$1 <= n \quad 1$
$3 <= n$

$(0, 1, 3, 6, 10, 15, 21, \ldots n)$

k - term

k^{th} term $= \frac{k + (k+1))}{2}$

$$n = \frac{k^2 + k}{2}$$

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

$$k^2 + k - 2n = 0$$

$$k = \frac{-1 + \sqrt{1^2 + 8n}}{2}$$

$$k = \frac{\sqrt{8n+1}}{2} - 1$$

$$k = \frac{\sqrt{8n+1}}{2} - \frac{\sqrt{8n}}{2} = \sqrt{n}$$

$T(n) = \sqrt{n}$ space $O(1)$

(6) void Recursion (int n) $\rightarrow T(n)$

```
{
    if (n= =1) return n;
    recursion (n-1)      -> T(n-1)
    print (n);           -> 1
    Recursion (n-1);     -> T(n-1)
}
```

$$T(n) = \begin{cases} 1 & n=1 \\ 2T(n-1)+1 & n>1 \end{cases}$$

$T(n) = 2T(n-1) + 1$ — ①

$T(n) = 2T_n(n-1) + 1$ — ①

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

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

$T(n) = 4T(n-2) + (1+2)$ — ②

$T(n-2) = 2(T(n-3)) + 1$

$T(n) = 2(T(n-3)) + 1$

$T(n) = 4(2T(n-3) + 1) + (1+2)$

$T(n) = 8(2T(n-4) + 1) + (1+2+4)$

$T(n) - 16T(n-4) + (1+2+4+8)$ — ④

$T(n) = 2^k T(n-k) + (1+2+4+8+ \cdots \cdots)$ k time

$T(n-k) = T(1)$

$k = n-1$;

$T(n) = 2^{n-1} \cdot T(1) + (1+2+4+8+ \cdots \cdots (n-1)$ time

$T(n) = \frac{2^n}{2} + (1+2+4+8 \cdots)$ $(n-1)$ time

$S_n = \frac{a(r^n - 1)}{r-1}$ $a = 1;\ r = 2.$ $n = n-1$

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

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

$T(n) = 2^{n-1}$

$\left(1 \cdot \frac{(2^{n-1} - 1)}{1}\right.$

$$T(n) = 2\left(\frac{2^n}{2}\right) - 1$$

$$T(n) = 2^n - 1$$

$$T(n) = O(2^n)$$

(7) It is a Binary Search algorithm

$$T(n) = \log_2(n)$$

$$T(n) = T\left(\frac{n}{2}\right) + 1$$

by using Master's Method (can't be solved)

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

So $a = 1$

$b = 2$

$f(n) = 1$

$$c = \log a = \log_2 1 = 0$$

$$n \cong f(n) = 1$$

$$n = f(n)$$

$$T(n) = O(\log_2 n)$$

8

$T(1) = 1$

1. $T(n) = T(n-1) + 1$ — ①

$T(n) = T(n-2) + 2$ — ②

$T(n) = T(n-3) + 3$ — ③

$$T(n) = T(n-k) + k$$
$$n - k = 1$$
$$k = n - 1$$
$$T(n) - T(1) + n - 1$$
$$= n$$
$$O(n)$$

2= $T(n) = T(n-1) + n$ — ①

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

$T(n) = T(n-2) + (n + (n-1))$ — ②

$T(n) = T(n-3) + (n + (n-1) + (n-2))$ — ③

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

$$T(n-k) = T(1)$$
$$n = k + 1$$
$$k = n - 1$$

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

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

$$T(n) = 1 + n\frac{(n+1)}{2}$$
$$= \frac{n^2 + n}{2} + 1$$
$$\frac{n^2 + n + 2}{2} \quad . \; O(n^2)$$

Ans 8

(3)

$$T(n) = T(n/2) + 1 \quad \text{--- (1)}$$

$$T(n/2) = T(n/4) + 1$$

$$T(n) = T(n/4) + 1$$

$$T(n/4) = T(n/8) + 1$$

$$T(n) = T\left(\frac{n}{8}\right) + 3 \quad \text{--- (3)}$$

$$T(n) = T\left(\frac{n}{2^k}\right) + k \quad \text{--- (4)}$$

$$\frac{n}{2^k} = 1$$

$$2^k = n$$

$$k = \log_2 n$$

$$T(n) = T(1) + \log_2(n)$$

$$T(n) = O(\log_2(n))$$

(4)

$$T(n) = 2T\left(\frac{n}{2}\right) + 1$$

$$c = 1$$

$$n^c = n$$

$$f(n) = 1$$

$$n^c > f(n)$$

$$T(n) = O(n)$$

vibhu gold PAGE NO DATE

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

$T(n) = O(2^n)$

(6) $T(n) = 3T(n-1)$; $T(0) = 1$

$T(n) = 3\,(T(n-1)) \cdot - O(1)$

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

$T(n) = 9T(n-2)$

$T(n) = 3^3\, T(n-3)$

$T(n) = 3^3 T(n-k)$

for $n - k = 0$

$n = k$

$T(n) = 3^n\,(0)$

$T(n) = 3^n$

$T(n) = O(3^3)$;

(7) $T(n) = \begin{cases} \cancel{\phi}\ 1 & ; n <= 2 \\ T(n) & ; n > 2 \end{cases}$

$T(n) = T(\sqrt{n}) + 1$ — ①

$T(\sqrt{n}) = T(n^{1/4}) + 1$

$T(n) = T(n^{1/4}) + 2$ — ②

$T(n) = T(n^{1/2^k}) + k$

for $T((\sqrt{n})^{1/k}) = T(2)$

$n^{1/k} = 2$

$n^{1/2k} = 2$

$\cancel{\frac{1}{2k}}\ \frac{1}{2^k} \log n = 1$

$2^k = \log n$; ~~$2^k \log$~~

$k = \log_2(\log(n))$

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

(8) $T(n) = T(\sqrt{n}) + n$

$T(\sqrt{n})^2 = T(n^{1/4}) + \sqrt{n}$

$T(n) = T(n^{1/4}) + n + \sqrt{n}$

$T(n) = T(n^{1/8}) + (n + \sqrt{n} + n^{1/4})$

$T(n) = T(n^{1/2k}) + \underbrace{(n + n^{1/2} + n^{1/4} + \cdots)}_{k\ \text{terms}}$

for $n^{1/2k} = 2$

$\frac{1}{2 \cdot k} = 1/\log(n)$

$k = \log(\log(n))$

$T(n) = 1 + (n + \sqrt{n} + \sqrt{n}\sqrt{n} + \cdots k \text{ time}$

$T(n) = 1 + \left(\begin{array}{l} G.P\ a = n \\ r = \sqrt{n} \\ \text{no of term} = k \end{array} \right)$

$T(n) = 1 + \left(\frac{n\,(\sqrt{n})^k - 1}{k - 1} \right)$

$T(n) = 1 + n \left(\frac{(\sqrt{n})^{\log\log(n)} - 1}{\log\log(n) - 1} \right)$

$T(n) = n \cdot \log\log(n)$ } neglectis other value

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

Ans (9) int sum = 0, i
for (i = 0 ; i < n ; i++)
{
sum += i)
}
0, 1, 2, ---- n .
So (T(n) = O(n) ; Space O(1)

(10) $O(N * (N, N-1, ----, 1))$
$O\left(N * \frac{(N+1)}{2}\right)$
$O(N * N)$

(11) $O\left(\frac{n}{2} * (\log_2 N)\right)$
$O(n \log N)$

(12) 2 (x) will always be a better choice for large input

(13) $O(\log N)$

(14) $T(n) = 7\left(T\left(\frac{n}{2}\right)\right) + (3n^2 + 2)$

$f(n) = 3n^2 + 2$
$a = 7$
$b = 2$
$C = \log_b a = \log_2 7$
$= 2.807$

$n^c = n^{2.8} \approx n^{2.8}$

$f(n) = 3n^2 + 2$

$n^c > f(n)$

$T(n) = O(n^{2.8})$

or $O(n^{2.8})$

$O(n^{2.8})$

$O(n^3)$

(15) $f_1(n) = n^{\sqrt{n}}$

$f_2(n) = 2^n$

$f_3(n) = (1.000001)^n$

$f_4(n) = n(10 \times 2^{n/2})$

o) $f_2(n) > f_4(n) > f_3(n) > f_1(n)$

(16) $f(n) = 2^{2n}$

$\log f(n) = 2n \log 2^2$

$\log f(n) = 2n$

or

$f(n) = 2^n \cdot 2^n$

$\Omega(2^n)$

(17) $T(n) = 2T\left(\frac{n}{2}\right) + n^2$

$c = 1$

$n^c = n$

$n^2 > n$

$f(n) > n^c$

$T(n) = O(n^2)$

(18) $O(\log N)$ (n keeps on decreasing by $n/2$)

(19) $T(n) = O(N^2 + N)$

$T(n) = O(N^2)$