Harshit Dhiman.

Tutorial-Sheet 2

DATE

PAGE

Sol(1) $i = 0, 1, 3, 6, 10, 15, 21, \ldots n.$

Let the sum of above K term is S_K.

$$S_K = 1+3+6+10+ \ldots + T_K \quad \text{—— (1)}$$
$$S_{K+1} = 1+3+6+10+ \ldots + T_{K+1} \quad \text{— (2)}$$

Subtract (2) from (1)

$$T_K = S_K - S_{K+1} = 1+2+3+4 \ldots + K$$

We have $T_K = n$

$$\therefore 1+2+3+4+ \ldots + K = n$$

$$\frac{K(K+1)}{2} = n \Rightarrow K^2 + K - 2n = 0$$

$$\Rightarrow K = \frac{-1 \pm \sqrt{8n+1}}{2}$$

taking only the value we get total no. of times the loop runs for $i = K+1 = \frac{\sqrt{8n-1}}{2}$

$$TC, T(n) = O\left(\frac{\sqrt{8n+1}}{2}\right)^2 = O(\sqrt{n})$$

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

$$T(n-1) \simeq T(n-2)$$
$$T(n) = 2T(n-2) + C$$
$$T(n-2) = 2 * (2T(n-2-2) + C) + C$$
$$= 4T(n-2) + 3C$$
$$T(n-4) = 2 * (4T(n-2) + 3C) + C$$
$$= 8T(n-3) + 7C$$

Generalizing

$$= 2^K T(n-K) + (2^K - 1)C$$

DATE _____
PAGE _____

put $n-k=0$

$n=k$.

put $n=k$

$$T(n) = 2^n * T(0) + (2^n-1)c$$
$$= 2^n * 1 + 2^n c - c$$
$$= 2^n(1+c) - c$$
$$= 2^n$$

Time complexity = $O(2^n)$

* Space complexity is proportional to the max. depth of recursion tree.

Hence space complexity of fibonacci recursion is $O(N)$.

SOL(3)

1. $n(\log n)$

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

2. n^3

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

DATE _____

PAGE _____

3. $\log(\log n)$

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

Soln (4) $T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + cn^2$

$\therefore T\left(\frac{n}{4}\right) \simeq T\left(\frac{n}{2}\right)$

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

As $a \geq 1$ and $b > 1$

Using master's method

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

$c = \log_b a$

$c = \log_2 2 = 1$

$f(n) > n^c$

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

$= O(n^2)$

Soln (5)

```
int fun(int)
for (int
```

for $i=1$, j is 1, 2, 3, 4 ---- runs for n times

for $i=2$, j is 1, 3, 5 ---- upto $n/2$ times

for $i=3$, j is 1, 4, 7 ---- upto $n/3$ times

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

DATE _____
PAGE _____

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

$$= n \int_1^n dx/x$$

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

$\Rightarrow$ Time complexity $= n \log n$.

Soln

for first iteration $i = 2$
second " $i = 2^K$
third " $i = (2^K)^K = 2^{K^2}$
⋮
nth iteration $i = 2^K$ loop end at $2^i = n$.

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

$K^i = \log n$.

$i = \log_e(\log n)$

Soln

99 to 1 in quick sort
whose pivot is where from front or end always.

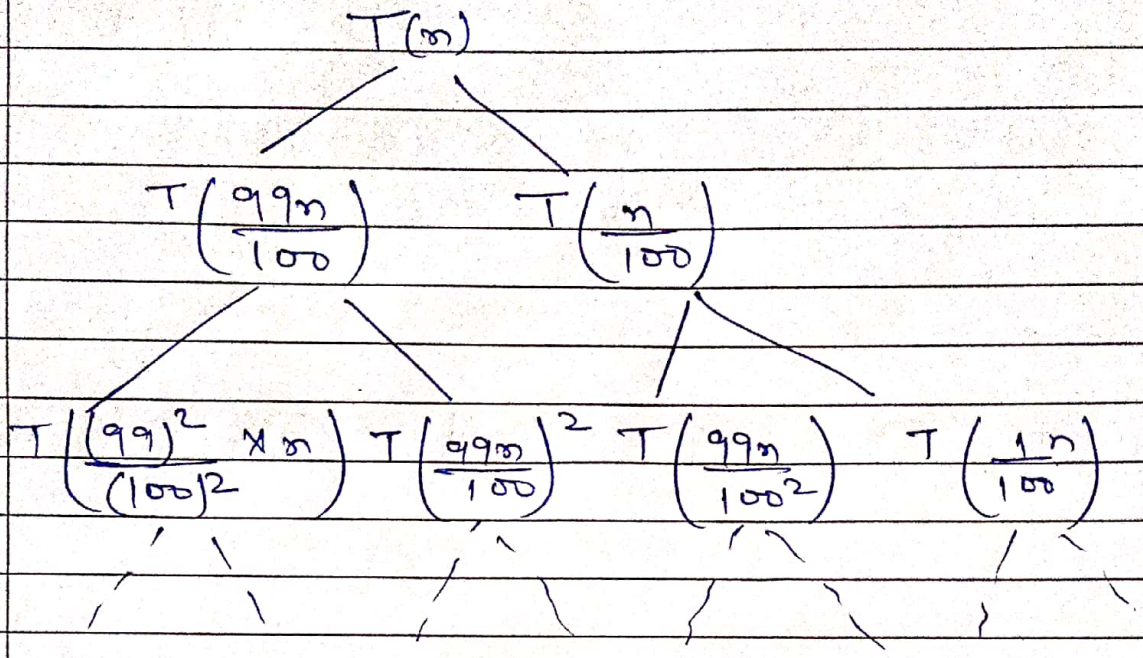
So,

$$T(n) = T\left(\frac{99n}{100}\right) + T\left(n/100\right) + O(n)$$

$$T(n) = T\left(\frac{99n}{100}\right) + T\left(\frac{n}{100}\right) + O(n)$$

DATE _____

PAGE _____



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

$$\log n = k \log 99/100$$

$$k = \log n \, \frac{100}{99}$$

$$TC = n * \log_{\frac{100}{99}}(n)$$

Soln 8 (a) $100 < \log\log(n) < \log^2 n < \log n < \log n! < n < n\log n < n^2 < 2^n < 4^n < 2^{(2^n)} < n^4$

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