Rümeysa Caylan
18291018

① From the lowest to the highest

$f_{12}(n) = 10^{15} < f_{14}(n) = \log\log n < f_{11}(n) = (\log n)^2 < f_{10}(n) = n^{1/3} + \log n <$

$f_2(n) = \sqrt{n} < f_3(n) = n - 1000, f_{13}(n) = n/\log n < f_4(n) = n\log n <$

$f_1(n) = n^2 + \log n < f_8(n) = n^{12} + n^{10} < f_9(n) = n^{12} \cdot \log n < f_7(n) = n^{11} \cdot 2^{2\log n} <$

$f_5(n) = 2^n + n^{10} < f_6 = n^5 + 3^n$

$f_3(n)$ and $f_{13}(n)$ are same order.

②

Basic operation is $\underline{r \leftarrow r+1}$

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{k=i+j-1}^{n} 1 = \sum_{i=1}^{n} \sum_{j=i+1}^{n} (n - (i+j+1) + 1)$$

$$= \sum_{i=1}^{n} \sum_{j=i+1}^{n} (n - i + 2) - \sum_{i=1}^{n} \sum_{j=i+1}^{n} j$$

$$\sum_{i=1}^{n} (n - i + 2) \underbrace{(n - (i+1) + 1)}_{(n-i)} - \sum_{i=1}^{n} \frac{n \cdot (n+1)}{2} - \frac{i(i+1)}{2}$$

$$= \frac{n^2 - n}{2} = O(n^2) \text{ times}$$

Algorithm returns $\frac{n^2 - n}{2}$

(3) $T(n) = \begin{cases} 1 & n \le 2 \\ 4T\left(\frac{n}{2}\right) + n^2 & n > 2 \end{cases}$

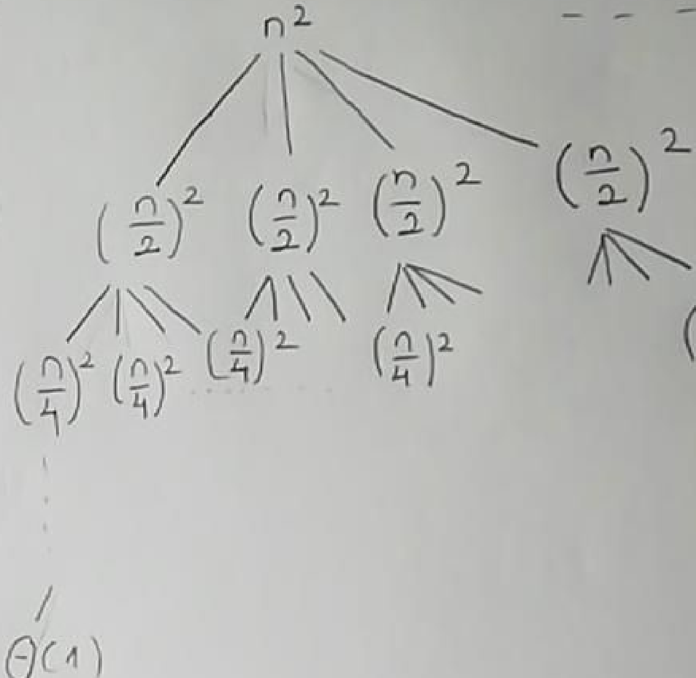
$T(n)$

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

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

$\vdots$

$T\left(\frac{n}{2^i}\right)$



n^2

$\frac{n^2}{\cancel{4}} \cancel{4}$ — n^2

$\frac{n^2}{\cancel{16}} \cancel{16}$ — n^2

$\vdots$

+

$n^2(1+1+1 \dots +1)$

$$\sum_{i=0}^{\log_2 n} n^2 = n^2 . \log_2 n + n^2$$

$$T(n) = \Theta(n^2 . \log_2 n)$$

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

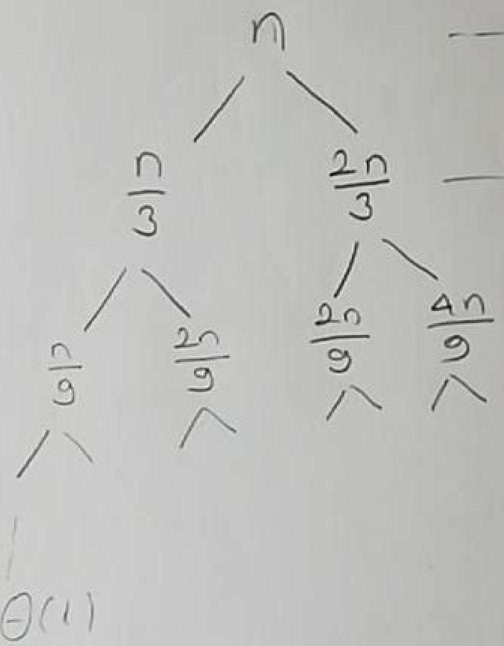
$\frac{n}{2^i} = 1$

$n = 2^i$

$\boxed{\log_2 n = i}$

(4) $T(n) = \begin{cases} 1 & n \leq 2 \\ T(\frac{n}{3}) + T(\frac{2n}{3}) + n & n > 2 \end{cases}$

$T(n)$



$\longrightarrow n$

$\longrightarrow \frac{3n}{3} = n$

$\longrightarrow \frac{9n}{9} = n$

$\vdots$

$+$ ______

$n + n + n \dots$

$$\sum_{i=0}^{\log_{3/2} n} n = (\log_{3/2} n + 1) \cdot n$$

$$= n \cdot \log_{3/2} n + n$$

$$T(n) \in \Theta(n \log_{3/2} n)$$

$$T\left(\frac{2^i n}{3^i}\right) = T(1)$$

$$\frac{2^i n}{3^i} = 1$$

$$n = \left(\frac{3}{2}\right)^i$$

$$\log_{3/2} n = i \longrightarrow 2^i = 2^{\log_{3/2} n} = n^{\log_{3/2} 2}$$

(5)

ALASKA $(A = (a_{ij})_{n\times n})$

```
r ← 0                          ——→ 1 operation
for i = 1 to n-2
    for j = i+1 to n
        if a_ij ≠ a_ji        ——→ 1 operation
            return false
return true
```

This algorithm computes is the matrix symmetric or not.

Basic operation is comparing the indices of the matrix.

$$\sum_{i=1}^{n-2}\sum_{j=i+1}^{n} 1 + 1 = \sum_{i=1}^{n-1} 1.(n-(i+1)+1) + 1$$

$$= \sum_{i=1}^{n-2} (n-i) = \sum_{i=1}^{n-2} n - \sum_{i=1}^{n-2} i + 1$$

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

$$= \frac{2n^2-4n-n^2+3n-2+2}{2}$$

$$= \frac{n^2}{2} - \frac{n}{2} = \underline{O(n^2) \text{ times}}$$

⑥ $T(n) = \begin{cases} 1 & n \leq 2 \\ 2T(n/2) + n\log n & n > 2 \end{cases}$

$a = 2 \qquad a \geq 1$ ✓

$b = 2 \qquad b > 1$ ✓

$f(n) = n\log n$

$$f(n) = O(n^{\log_2 2} \cdot \log^k n) \rightarrow k = 1$$

$$= O(n^{\log_2 2} \cdot \log n)$$

$$T(n) = \Theta(n^{\log_2 2} \cdot \log^2 n) = \Theta(n \cdot \log^2 n)$$

⑦ $T(n) = \begin{cases} 1 & n \leq 2 \\ 3T(n/3) + \sqrt{n} & n > 2 \end{cases}$

$a = 3 \qquad a \geq 1$ ✓

$b = 3 \qquad b > 1$ ✓

$\boxed{f(n) = \sqrt{n}}$

$$f(n) = O(n^{\log_3 3 - \varepsilon})$$

$$= O(n^{1-\varepsilon}) = n^{1/2} \qquad \varepsilon \rightarrow 1/2$$

$$T(n) = \Theta(n^{\log_2 2}) = \Theta(n)$$

(8) This algorithm computes the smallest element in the sequence.
It splits the array in the middle and algorithm works recursively until finds to smallest number.

$T(n) = 2T(n/2) + c\,\Theta(1)$ $\quad T(n) = 0$ for $n = 1$

$T(n) = 2[2T(n/4) + c] + c = 2^2 . T(n/2^2) + 2c + c$

$T(n) = 2^2 . [2T(n/8) + c] + c = 2^3 . T(n/2^3) + 4c + c$

$T(n) = 2^i . [T(n/2^i)] + (2^i + 1)c$

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

$T(n) = 2^i T(n/2^i) + (2^i + 1) . c$

$n = 2^i$

$= 2^{\log_2 n} T(n/2^{\log_2 n}) + (2^{\log_2 n} + 1) . c$

$\boxed{\log_2 n = i}$

$= n . T(1) + c . (n+1)$

$= O(cn) \Rightarrow \Theta(n)$