$\log n < n^{1/2} < n < n\log n < n^2 < 2^n < n!$

1) × $f_1 \rightarrow O(n^2) \rightarrow$ quadratic

× $f_2 \rightarrow O(\frac{n}{\log n})$

× $f_3 \rightarrow O(n) \rightarrow$ linear

× $f_4 \rightarrow O(n\log n) \rightarrow$ linearithmic

/ $f_5(n) = n^8 \cdot 2^{2\log n}$

$= n^8 \cdot 2^{\log n^2}$

$= n^8 \cdot n^{\log 2}$

$= n^{10} = O(n^{10})$

↳ polynomial

× $f_6 \rightarrow O(\log\log n)$

/ $f_7 \rightarrow O(3^n) \rightarrow$ exponential

$f_8 \rightarrow O(n^{10}\log n)$

/ $f_9 \rightarrow O(n^{10}) \rightarrow$ polynomial

× $f_{10} \rightarrow O(n^{1/2}) \rightarrow$ polynomial

× $f_{11} \rightarrow O(\log^2 n)$

× $f_{12} \rightarrow O(10^{15}) \rightarrow$ constant

× $f_{13} \rightarrow O(n^{1/2}) \rightarrow$ polynomial

$f_{14} \rightarrow O(5^n) \rightarrow$ exponential

$f_{12} < f_6 < f_{11} < f_{10} < f_{13} < f_2 < f_3 < f_4 < f_1 < f_5 = f_9 < f_8 < f_7 < f_{14}$

2)

$r \leftarrow 1$ — operation 1

for $i = 1$ to n → This loop iters n times: $O(n)$

for $j = i$ to n → This loop iters $n-i+1$ times : $O(n)$

for $k = j$ to n → This loop iters $n-j+1$ times : $O(n)$

$r \leftarrow r + r$ → $O(1)$

operation 2

return r

operation 3

$$\sum_{i=1}^{n} \sum_{j=i}^{n} \sum_{k=j}^{n} 1$$

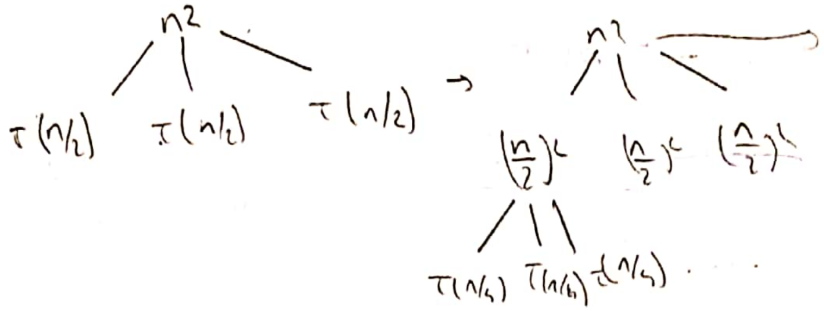
$$\sum_{i=1}^{n} \sum_{j=i}^{n} n-j$$

$$\sum_{i=1}^{n} (n-j)\cdot(n-i) = \underbrace{n\cdot(n-j)\cdot(n-i)}$$

The operation 2 executed $n\cdot(n-j)\cdot(n-i)$ times

worst case running time is $O(n^3)$

3) $T(n) = \begin{cases} 1, & \text{if } n \le 2 \\ 3T(n/2) + n^2, & \text{if } n > 2 \end{cases}$



n^2

$\frac{3}{4} n^2$

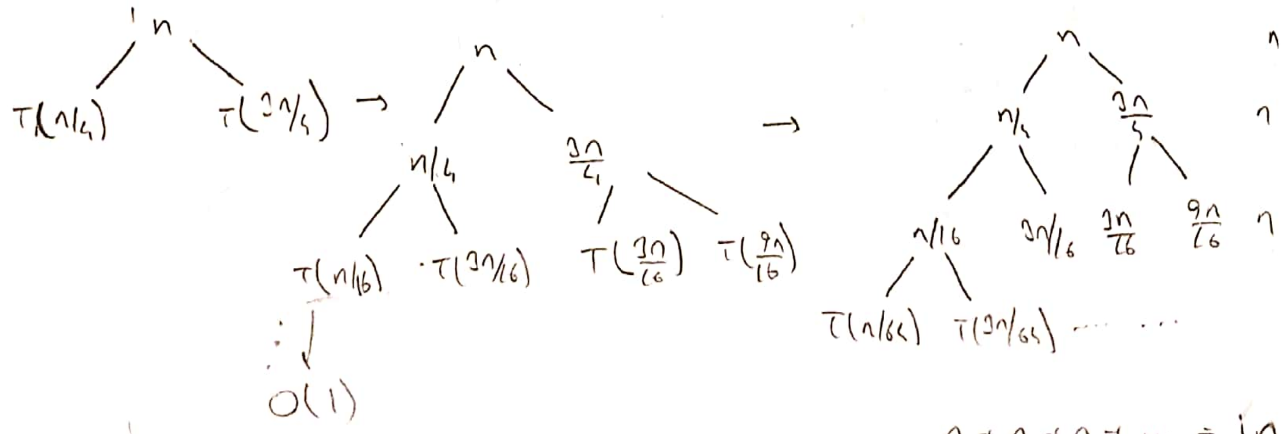
$\frac{9}{16} n^2$

$\vdots$

$n^2 \underbrace{\left(1 + \frac{3}{4} + \left(\frac{3}{4}\right)^2 + \dots\right)}_{\frac{1}{1-\frac{3}{4}} = \frac{1}{\frac{1}{4}} = 4}$

$4n^2 \to O(n^2)$

4) $T(n) = \begin{cases} 1, & \text{if } n \le 2 \\ T(n/4) + T(3n/4) + n, & \text{if } n > 2 \end{cases}$



$\underbrace{n + n + n + \dots}_{i} = in$

$n/4^i = 1$

$n = 4^i$

$i = \log_4 n$

$\log_4 n \cdot n = O(n \log n)$

6) $T(n) = \begin{cases} 1 & , \text{ if } n \leq 2 \\ 9T\left(\frac{n}{3}\right) + n^2 \log n & , \text{ if } n > 2 \end{cases}$

$a \geq 1$ $b > 1$ non-negative

$n^{\log_3 9} = n^2$? $f(n) = n^2 \log n$

$\therefore T(n) = n^2 \log^2 n$

7) $T(n) = \begin{cases} 1 & , \text{ if } n \leq 2 \\ 5T(n/5) + \sqrt{n} & , \text{ if } n > 2 \end{cases}$

$a \geq 1$ $b > 1$ non-negative

$n^{\log_5 5} = n$? $\sqrt{n}$

$T(n) = n$

5) $r \leftarrow 0$

for $i = 1$ to $n-2$

for $j = i+1$ to n

if $a_{ij} \neq a_{ji}$

return false

return true

$n-i+1$ times

$n-2$ times

Basic operations

$r \leftarrow 0$

comparison between a_{ij} and a_{ji}

return statements.

$$\sum_{i=1}^{n-2} \sum_{j=i+1}^{n} 1 = \sum_{i=1}^{n-2} (n-i+1) = (n-2).(n-i+1) \text{ times basic operation executed}$$

Time complexity $O(n^2)$

Algorithm checks if matrix is symmetric

$$\text{Antalya}(\overbrace{a_i, a_{i+1}, \dots, a_j}^{n})$$

8) input: a sequence of integers

if $i = j$

return a_i

else

mid $\leftarrow \lfloor (i+j)/2 \rfloor$

temp1 $\leftarrow$ Antalya $(a_i, \dots, a_{mid})$

temp2 $\leftarrow$ Antalya $(a_{mid}, \dots, a_j)$

if temp1 $\geq$ temp2

return temp1

else

return temp2

Finds the min of array,

$$T(n) = \begin{cases} 1, & \text{if } n \leq 1 \\ 2T(n/2) + x, & \text{if } n > 1 \end{cases}$$

$$T(n) = 2T(n/2) + x$$

$$= 2[2T(n/4) + x] + x = 2^2 T(n/2^2) + 3x$$

$$= 2(2[2T(n/8) + x] + x) + x = 2^3 T(n/2^3) + 7x$$

$$T(n) = 2^i T(n/2^i) + x.(2^{i-1} + 2^{i-2} + \dots + 2^0)$$

to find $i \Rightarrow \frac{n}{2^i} = 1$

$$2^i = n$$

$$i = \log_2 n$$

$$T(n) = 2^{\log_2 n} . T(1) + x.(2^{\log_2 n - 1} + 2^{\log_2 n - 2} + \dots + 2^0)$$

$$= n + x\left(\frac{2^{\log n}}{2} + \dots\right)$$

$$= n + x.\frac{n}{2} = O(n)$$