// البسيط الشكل //

Büyük O - notasyon: iki tane çalışma zamanı verilsin $f(n)$ ve $g(n)$

$f(n) = O(g(n))$ eğer bir sabit c sayısı ve bir pozitif int n_0 var ise ve:

$$\boxed{f(n) \leq c\, g(n)} \quad \text{tüm} \quad n \geq n_0$$

// Big-O: Upper bound. // üst Limit.

örnek: $\frac{1}{2}n^2 + 3n \in O(n^2)$?

(üstte: $f(n)$, $g(n)$)

bunun üst Limiti $\xrightarrow{\text{Bu mı?}}$ $O(n^2)$

Bunun değe gösterebilmek için c ve n_0

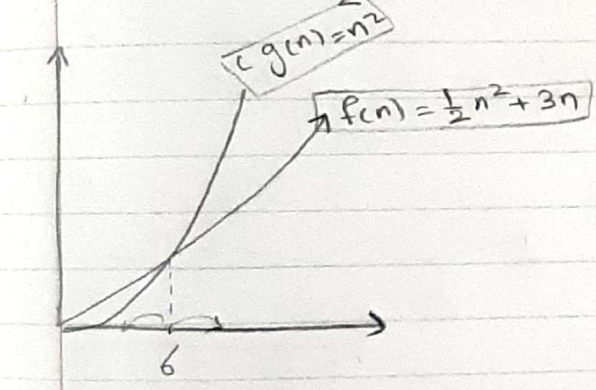
$$0 \leq \frac{1}{2}n^2 + 3n \leq cn^2$$

$c = 1$

$$\frac{1}{2}n^2 + 3n \leq n^2 \Rightarrow 3n \leq \frac{1}{2}n^2 \Rightarrow \boxed{6 < n}$$

$\boxed{c = 1}$

$\boxed{n_0 = 6}$



6: $\frac{1}{2}.(6)^2 + 3(6) = \boxed{36}$

$f(n)$ $\quad$ $g(n)$

8: $g(n) = \boxed{64}$

$f(n) = \frac{1}{2}64 + 24 = 32 + 24 = \boxed{56}$

4: $g(n) = \boxed{16}$

$f(n) = \frac{1}{2}16 + 3.(4) = 8 + 12 = \boxed{20}$

// Big-Omega (Ω): Lower Bound. en az Limit //

$f(n)$ fonksiyonumuz var $\Omega(g(n))$ göstermek için bir $\boxed{c}$ sabiti ve pozitif $\boxed{n_0}$ integer olmalı öyle ki:

$$\boxed{c \cdot g(n) \leq f(n)}$$

örnek: $n^2 + 10n \in \Omega(n^2)$ $\quad n_0 \geq 0 \quad c = 1$

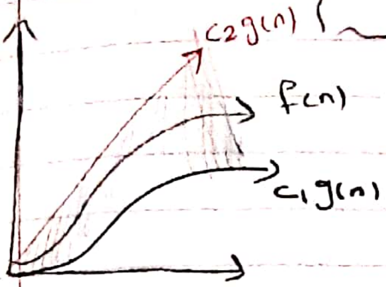
$c(n^2) \leq n^2 + 10n$

$\boxed{n^2 \leq n^2 + 10n}$ ✓

Big-Theta Θ : $f(n)$ ve $g(n)$ $f(n) \in \Theta(g(n))$

Eğer $\boxed{c_1 \text{ ve } c_2}$ sabitleri, birde pozitif $\boxed{n_0}$ varise

$$\boxed{c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)}$$



Örnek: $\underbrace{\frac{1}{2}n^2 - 3n}_{f(n)} = \Theta(\underbrace{n^2}_{g(n)})$ $c_1 =$ $c_2 =$

c_1 ve c_2 ve n_0

$0 \leq c_1 n^2 \leq \frac{1}{2}n^2 - 3n \leq c_2 n^2$ tüm $n \geq n_0$

$c_1 \leq \frac{1}{2} - \frac{3}{n} \leq c_2$

$c_1 \leq \frac{1}{2} - \frac{3}{7} \Rightarrow$ $\frac{1}{2} - \frac{3}{n} \leq c_2$

$\boxed{c_1 \leq \frac{1}{14}}$ $\boxed{\frac{1}{2} \leq c_2}$

$n_0 = 7$

$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$

$\frac{1}{14} \cdot n^2 \leq \frac{1}{2}n^2 - 3n \leq \frac{1}{2} \cdot n^2 \Rightarrow$ 7 için

$\Rightarrow \boxed{\frac{49}{14} \leq \frac{1}{2} \cdot 49 - 21 \leq \frac{49}{2}}$

$\Rightarrow f(n) = \Theta(g(n))$

Best Case, Worst Case, Avrage Case:

Ψ_n : tüm inputlar.

$T(I)$: temel işlemler.

Best Case:

$$B(n) = \min \{ T(I) \in \Psi_n \}$$

10	15	20	78	90

↑ إذا كنت أبحث عن 20 مثلاً

Worst Case:

Ptr → [ ] → [ ] → [ ] → [ ] → [ ]

$$W(n) = Ma \{ T(I) \in \Psi_n \}$$

Avrage Case:

$$A(n) = \sum T(I) . p(I)$$

0 1 2 3 4 6 7 ← 8 boyutlu

sıfır arıyorum mesela → 3 adımla.

3 2 3 1 3 2 3 4

$\frac{21}{8} < \log_2 8$

عملية ترتيب

insertion sort (A):

A sort in incressing order

Const / Cost — time / Time

for $J \leftarrow 2$ to Lenght [A]	c_1	$(n-1+1)$
do key ← A[J]	c_2	$n-1$
A Insert A[J] And A[J] > key	c_3	$n-1$
$i \leftarrow (J-1)$	c_4	$\sum_{j=2}^{n} t_c$
while $i > 0$ and $A[J] > key$		
do $A[i+1] \leftarrow A[i]$	c_5	$\sum_{j=2}^{n} (t_c - 1)$
$i \leftarrow i-1$	c_6	
$A[i+1] \leftarrow key$.	c_6	$n-1$

Sıralanmış dizi ise $\boxed{t_c = 1}$

$$t(n) = n c_1 + (n-1) c_2 + (n-1) c_3 + c_4 \left(\sum_{j=2}^{n} t_c \right) + c_5 \left(\sum_{j=2}^{n} (t_c - 1) \right) + c_6 (n-1)$$

$= \text{Best Case} = O(n)$

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

$$T(n) = c_1 * n + c_2 * (n-1) + c_4 \cdot (n-1) + c_5 \cdot (\sum_{j=2}^{n} t_c) +$$
$$c_6 * \sum_{j=2}^{n} (t_c - 1) + c_7 * \sum_{j=2}^{n} (t_c - 1) + c_8 * (n-1)$$

$$t(n) = c_1 * n + c_2 * (n-1) + c_4 * (n-1) + c_5 * (n-1) +$$
$$c_8 \cdot (n-1)$$

Warst case:

$t_j = j \qquad j = 2, 3, \ldots n$

$$\sum_{j=2}^{n} j = \frac{n \cdot (n+1)}{2} - 1 \qquad \sum_{j=2}^{n} j - 1 = \frac{n \cdot (n+1)}{2}$$

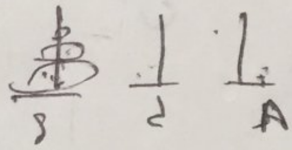
$$t_n = c_1 \cdot n + c_2 (n-1) + c_4 * (n-1) + c_5 * \left(\frac{n * (n-1)}{2}) - 1 \right) +$$
$$c_6 \cdot \left(\frac{n \cdot (n-1)}{2} \right) + c_7 \cdot \left(\frac{n * (n-1)}{2} \right) + c_8 (n-1)$$
$$= c_1 * n + c_2 * n - c_2 + c_4 * n - c_4 + c_5 \frac{n^2}{2} + c_5 \frac{n}{2} -$$
$$c_5 + c_6 \frac{n^2}{n} - \frac{c_6 * n}{2} + c_7 * \frac{n^2}{2} - c_7 * \frac{n}{2} + c_8 * n - c_8$$
$$n^2 \left(\frac{c_2}{2} + \frac{c_6}{2} + \frac{c_7}{2} \right) + n \left(c_1 + c_2 + \frac{c_5}{2} - \frac{c_6}{2} - \frac{c_7}{2} + c_8 \right.$$

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

Recursive algoritmaları çözerken takip ettiğimiz yöntemler

20.10.2020 Hafta-3

1) İterasyon yöntemi
2) Değiştirme (substitution) yöntem
3) Master yöntemi
4) Karakteristik denklem yöntemi.



---

1) İterasyon yöntemi: Yinelemeleri toplama dönüştürüp sınırlandırma yapar.

Example: $t_n = 2t_{n-1} + 1$ $\quad n \geq 2 \quad t_1 = 1$

$\Downarrow$

```
void hanoi(int n, char source, char dest, char spare)
{  if (n>0)
     hanoi(n-1, source, spare, dest);
     cout << "move top disk from pole" << source << " to pole" << dest <<
     hanoi(n-1, spare, dest, source);
}
```

$t_{n-1} = 2t_{n-2} + 1$

Recurrence: Bir fonksiyon yine kendi türünden edilebiliyorsa o fonksiyon recursive.

Cost

```
int factorial(int n)         c1
{
    if (n>1)                 c2
        return n*factorial(n-1)   c3(?)
    else
        return 1;            c4
}.
```

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

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

$$= c + c + c + T(n-3)$$

$$= c + c + c + \dots + T(1)$$

CamScanner ile tarandı

ık: $t_n = t_{n-1} + n \qquad n \geq 2, \quad t_1 = 1.$

$t_{n-1} = (n-1) + t_{n-2}$

$t_{n-2} = (n-2) + t_{n-3}$

$\vdots$

$t_{n-3} = t_{n-4} + n-3$

$\vdots$

$\cancel{t_2 = t_1 +}$

$t_n \in \Theta\left(\frac{n^2}{2}\right)$

$\boxed{t_n \in \Theta(n^2)}$

$t_{n-1} = t_{n-2} + n-1$

$t_{n-2} = t_{n-3} + n-2$

$t_n = t_{n-2} + n-1 + n$

$t_n = (t_{n-3} + n-2) + n-1 + n$

$t_n = t_{n-3} + n-2 + n-1 + n$

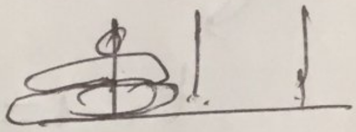
$= \underbrace{t_{n-(n-1)}}_{t_1 = 1} + \cancel{n} - (\cancel{n}-2) + \cancel{n} - (\cancel{n}-3) + \dots n$

$t_n = 1 + 2 + 3 + \dots + n = \frac{n.(n+1)}{2}$

$$t_n = 2t_{n-1} + 1 \qquad n \geq 2 \qquad t_1 = 1$$

$$\boxed{t_{n-1} = 2t_{n-2} + 1}$$

$$\boxed{t_{n-2} = 2t_{n-3} + 1}$$



$$\vdots$$

$$t_n = 2\left(2\cdot\left(2t_{n-3} + 1\right) + 1\right) + 1$$

$$t_n = 2^3 t_{n-3} + 2^2 + 2 + 1$$

$$= 2^{n-1} t_{n-(n-1)} + 2^{n-2} + \dots + 2 + 1$$

$$= 2^{n-1} \cdot t_1 + 2^{n-2} + \dots + 2 + 1$$

$$= 2^{n-1} + 2^{n-2} + \dots + 2 + 1$$

$$= 2^n - 1$$

$$\boxed{t_n \in O(2^n)}$$

Substitution (guess) method:

Sınır tahmini yapıldıktan sonra matematiksel tümevarım yöntemiyle tahminin doğruluğu kanıtlanır.

örnek:

$$t_n = t_{n/2} + c \qquad n \geq 2, \quad t_1 = 1$$

$$t_2 = t_1 + c = 1 + c$$

$$t_4 = t_2 + c = 1 + c + c$$

$$t_8 = t_4 + c = 1 + c + c + c$$

$$\vdots$$

$$t_8 = 1 + 3c$$

$$t_{2^k} = 1 + k.c \qquad \boxed{n = 2^k}$$

$$\boxed{t_n = 1 + c.\log n} \qquad \boxed{t_n \in \Theta(\log n)}$$

Tümevarım yöntemiyle ispatı:

Base case: $n = 1 \qquad t_1 = 1$ ✓

Induction case: $n = 2^k, \quad t_{2^k} = 1 + k.c$ doğru

$$t_{2^k} = t_{2^k/2} + c$$

Proof case: $n = 2^{k+1}, \quad t_{2^{k+1}} = 1 + (k+1).c = 1 + kc + c$

$$t_{2^{k+1}} = t_{2^{k+1}/2} + c = t_{2^k} + c = \underbrace{1 + kc}\, + c$$

Örnek:

$$t_n = n + 2t_{(n/2)}$$

$n \geq 2 \qquad t_1 = 0$

$$t_n = 2t_{n/2} + n$$

$$t_n = 2 \cdot t_{n/2} + n$$

$$t_{n/2} = 2\, t_{n/4} + n/2$$

$$t_{n/4} = 2 \cdot t_{n/8} + \frac{n}{4}$$

$\vdots$

$$t_n = 2\left(2 \cdot \left(2\, t_{n/8} + \frac{n}{4}\right) + \frac{n}{2}\right) + n$$

$$= 2^3 \cdot t_{n/8} + \cancel{2}^{\cancel{2}} \cdot \frac{n}{\cancel{4}} + \cancel{2} \cdot \frac{n}{\cancel{2}} + n$$

$$= 2^3 \cdot t_{n/2^3} + n + n + n$$

$$= 2^3 \cdot t_{n/2^3} + 3 \cdot n$$

$$= n \cdot t_{\frac{n}{n}} + k \cdot n$$

$$= \cancel{n \cdot 0} + k \cdot n$$

$$(\lg n) \cdot n \sim n \lg n$$

$$\boxed{t_n \in \Theta(n \lg n)}$$

$n = 2^k \Rightarrow k = ?$

$\lg n = \lg 2^k$

$\lg n = k \cdot \lg \cancel{2}$

$\boxed{k = \lg n}$

CamScanner ile tarandı

Master method:

$T(n) = a\,T(n/b) + f(n)$ şekline

dönüştürüp, $a \geq 1$ $b \geq 2$ kurallarımız

var.

$$T(n) = a\,T\left(\frac{n}{b}\right) + c \cdot n^{i}$$

$$T_n \in \begin{cases} \Theta(n^{i} \log_b n) & \text{case 1: } a = b^{i},\ i = \log_b a \\ \Theta(n^{\log_b a}) & \text{case 2: } a > b^{i},\ i < \log_b a \\ \Theta(n^{i}) & \text{case 3: } a < b^{i},\ i > \log_b a. \end{cases}$$

örnek: $t_n = 2t_{n/2} + n$ $n \geq 2$ $t_1 = 1$

$a = 2$, $b = 2$, $i = 1$

case 1: $a = b^{i} \Rightarrow 2 = 2^{1}$ ✓

$\Theta(n \lg n)$

örnek: $t_n = 8t_{n/2} + c \cdot n^2$

$a = 8$, $b = 2$, $i = 2$

$a > b^{i}$, $8 > 2^2$

$\Theta(n^{\log_2 8})$

$\boxed{\Theta(n^3)}$

Örnek:

$t_n = t_{n/2} + n \qquad n \geq 2, \quad t_1 = 0$

$a = 1 \quad , \quad b = 2 \quad , \quad i = 1$

Case-3: $a < b^i$

$1 < 2^1$

$\Theta(n^1) = \Theta(n)$ ✓

CamScanner ile tarandı

4. Karekteristik Denklem ile Çözme: $t_n = t_{n-1} + 1$ | t. haf ba.

$$a_0 t_n + a_1 t_{n-1} + \dots + a_k t_{n-k} = 0$$

1) $t_i \cdot t_{i+j}$ veya t^2 ifadeler içermediğinde.
2) a_i katsayıları sabit olduğunda.
3) Lineer combinasyonları sıfıra eşit olduğundan bu denklem bir homojen denklemdir.

$$t_n = t_{n-1} + t_{n-2}$$
$$t_n - t_{n-1} - t_{n-2} = 0$$

$$a_0 t_n + a_1 t_{n-1} + a_2 t_{n-2} + \dots + a_k t_{n-k} = 0 \quad \dots (1)$$

$t_n = x^n$

$$a_0 x^n + a_1 x^{n-1} + \dots + a_k \cdot t^{n-k} = 0$$

$x=0$ geçici çözüm, bu bizi ilgilendirmiyor.

$$a_0 \cdot x^k + a_1 \cdot x^{k-1} + \dots + a_k = 0$$

Bu iki dereceden denklem (1) nolu denklemin karekteristik denklemdir.

=//=

Polinom çözümünde iki dereceden bir ifadenin k adet kökü (farklı olmak zorunda değildir).

$$p(x) = \prod_{1 \le i \le k} (x - r_i)$$

$p(r_i) = 0$ $x = r_i$ olmasına gelmektedir.

Recursive ifadenin çözümünde karekteristik denklem r_i^n olmasını gerektir.

* Çözümlerin Linner kompinasyonu yine çözümü doğruluğunda. Her hangi $c_1, c_2, \dots, c_k$ katsayı için

$$\boxed{t_n = \sum c_i \cdot r_i^n}$$

* Eğer k. dereceden denklemimiz var ise (k) adet başlangıç sabitiniz olmalı.

=//=

Ex $t_n = 3t_{n-1} + 4t_{n-2}$ $\quad n \geq 2 \quad t_0 = 0, \; t_1 = 1$

$t_n - 3t_{n-1} - 4t_{n-2} = 0$

$1x^k - 3x^{k-1} - 4 \cdot x^{k-2} = 0$ ikinci

$x^2 - 3x - 4 = 0$

$x \quad -4$

$x \quad +1$ $\quad \boxed{r_1 = 4}$

$(x-4)(x+1) = 0$ $\quad \boxed{r_2 = -1}$

$t_n = \sum c_i \cdot r_i^n$

$t_n = c_1 \cdot (4)^n + c_2 \cdot (-1)^n$

$t_0 = c_1 \cdot (4)^0 + c_2 (-1)^0 = c_1 \cdot 1 + c_2 \cdot 1 = 0$

$t_1 = c_1 \cdot (4)^1 + c_2 (-1)^1 = 4c_1 + (-1) \cdot c_2 = 1$

$c_1 = -c_2$

$4(-c_2) - c_2 = 1 \Rightarrow -5c_2 = 1 \Rightarrow \boxed{c_2 = -\frac{1}{5}}, \quad \boxed{c_1 = \frac{1}{5}}$

$t_n = \sum c_i \cdot r_i^n = \left(\frac{1}{5}\right) \cdot (4)^n + \left(-\frac{1}{5}\right) \cdot (-1)^n$

$\Rightarrow \boxed{t_n \in \Theta(4^n)}$

=//=

$a_1 = \frac{-b \mp \sqrt{\Delta}}{2a} = \frac{-1 + \sqrt{5}}{2}$ $b^2 - 4ac$ x

Ex fibonacci karmaşıklığı $\Rightarrow \frac{-1-\sqrt{5}}{2}$ $1 - 4 \cdot 1 \cdot (-1) = 1 + 5 = 6\sqrt{5}$

$t_n = t_{n-1} + t_{n-2} \quad n \geq 2 \qquad t_0 = 0,\ t_1 = 1$

$t_n - t_{n-1} - t_{n-2} = 0$

$x^k - x^{k-1} - x^{k-2} = 0 \quad (k=2)$

$(x^2 - x - 1) = 0$

$\boxed{r_1 = \frac{1+\sqrt{5}}{2}} \qquad \boxed{r_2 = \frac{-1+\sqrt{5}}{2}}$

$t_n = \sum c_i \cdot (r_i)^n$

$= c_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + c_2 \cdot \left(\frac{-1+\sqrt{5}}{2}\right)^n$

$\Rightarrow$

$t_0 = c_1 \cdot \left(\frac{1+\sqrt{5}}{2}\right)^0 + c_2 \cdot \left(\frac{-1+\sqrt{5}}{2}\right)^0 = 0$

$\Rightarrow \{c_1 + c_2 = 0\}$

$=$

$t_1 = c_1 \cdot \left(\frac{1+\sqrt{5}}{2}\right)^1 + c_2 \left(\frac{-1+\sqrt{5}}{2}\right)^1 = 1$

$\Rightarrow c_1 = \frac{1}{\sqrt{5}}, \quad c_2 = \frac{-1}{\sqrt{5}}$

$t_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$

$\Rightarrow t_n \in \Theta\left(\frac{1+\sqrt{5}}{2}\right)^n$

Bulduğumuz köklerin aynı olma durumu

- tekrar eden kökler için:

$t_n = n \cdot r^n$

Bu da

$a_0 \cdot n \cdot r^n + a (n-1) \cdot r^{n-1} + \dots \ a_k (n-k) \cdot r^{n-k} = 0$

$t_n = r^n, \quad t_n = n \cdot r^n, \quad t_n = n^2 \cdot r^n, \dots$

$t_n = (r_1 \cdot c_1 + r_2 \cdot c_2 + n \cdot c_3)$

Ex: $t_n = 5t_{n-1} - 8t_{n-2} + 4t_{n-3}$ $\quad n \geq 3$

$t_0 = 0, \quad t_1 = 1, \quad t_2 = 2$

$t_n - 5t_{n-1} + 8t_{n-2} - 4t_{n-3} = 0$

$x^n - 5x^{k-1} + 8x^{k-2} - 4x^{k-3} = 0$

$x^3 - 5x^2 + 8x - 4 = 0$

$(x-1)\cdot(x-2)^2 = 0$

$\begin{cases} r_1 = 1 \\ r_2 = 2 \\ r_3 = 2 \end{cases}$

$t_n = \sum c_i \cdot r_i^n \Rightarrow c_1 \cdot (1)^n + c_2 \cdot (2)^n + c_3 \cdot (2)^n \cdot n$

$n=0 \Rightarrow 0 = c_1 + c_2 + 0 \Rightarrow$

$n=1 \Rightarrow 1 = c_1 + 2c_2 + 2c_3$

$n=2 \Rightarrow 2 = c_1 + 4c_2 + 8c_3$

$$\left[\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 1 & 2 & 2 & 1 \\ 1 & 4 & 8 & 2 \end{array}\right]$$

$$\xrightarrow{-c_1} \left[\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 3 & 8 & 2 \end{array}\right] \longrightarrow \left[\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 2 & -1 \end{array}\right] \rightarrow \left[\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & -\frac{1}{2} \end{array}\right]$$

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & -1/2 \end{array}\right] \quad \begin{cases} c_1 = -2 \\ c_2 = 2 \\ c_3 = -1/2 \end{cases}$$

$t_n = 2 \cdot (1)^n + 2 \cdot (2)^n - \frac{1}{\cancel{2}} \cdot (n)\ (2)^n$

$t_n \in \Theta(n \cdot (2)^n)$

Homojen olmayan denklemlerdir.

$a_0 t_n + a_1 t_{n-1} + a_2 + t_{n-2} + \cdots + a_k t_{n-k} = b^n p(n)$

$\Downarrow$

Bildiğimiz homojen forma dönüştürüyoruz.

$\Downarrow$

$(a_0 . x^k + a_1 . x^{k-1} + \cdots a_k)(x-b)^{d+1} = 0$ form

b: sabit sayı

d: polinomun derecesi

Ex $t_n = 2t_{n-1} + 1$, $n \geqslant 1$ ve $t_0 = 1$

$t_n - 2t_{n-1} = 1 \rightarrow b = 1$ $\quad p(n) = 1$

$(x-2)(x-1)^2 = 0$ $\quad r_1 = 2$, $r_2 = 1$

$t_n = c_1 . (1) + c_2 . 2^n$

$n=0 \Rightarrow 0 = \boxed{c_1 + c_2 = 1}$

$n=1 \Rightarrow t_1 = c_1 + 2c_2$

$2t_0 + 1 = c_1 + 2^1 c_2$

$\Rightarrow 0 + 1 = c_1 + 2c_2 \Rightarrow \boxed{c_1 + 2c_2 = 1}$

$\begin{cases} c_1 = -1 \\ c_2 = 1 \end{cases}$

$t_n = (-1) . (1)^n + 1 2^n \rightsquigarrow t_n \in \Theta(2^n)$

Ex

$$a_0 t_n + a_1 . t_{n-1} + \cdots + a_k . t_{n-k} = b_1^n P_1(n) + b_2^n (n) + \cdots$$

$\Downarrow$

$$(a_0 x^k + a_1 . x^{k-1} + \cdots + a_k)(x - b_1)^{d_1+1} . (x - b_2)^{d_2+1} \cdots = 0$$

Ex

$t_n = 2t_{n-1} + n + 2^n \qquad n \geq 1 \quad , \quad t_0 = 0$

$t_n - 2t_{n-1} = \underline{n} + 2^n$

$b_1 : 1 \qquad b_2 = 2$

$d : 1 \qquad P_2(n) ; 1$

$p(n) ; n \qquad d_2 : 0$

$(x-2).(x-1)^2 . (x-2)^1 = 0 \quad = 0$

$r_1 = 2 . \quad r_2 = 1 \quad r_4 = 2$

$r_3 = 1$

$r_1 = 1$
$r_2 = 1$
$r_3 = 2$
$r_4 = 2$

$t_n = C_1 . 2^n + C_2 . (1)^n + C_3 n (1)^n + C_4 . n (2)^n$ X

$\Rightarrow \quad C_1 + C_2 + 0C_3 + 0C_4 = 0 \Rightarrow \boxed{C_1 + C_2 = 0}$

$1 \Rightarrow \quad 2C_1 + C_2 + C_3 + 2C_4 = 3$

$\Rightarrow \quad 4C_1 + C_2 + 2C_3 + 8C_4 = 12$

$\Rightarrow \quad 8C_1 + C_2 + 3C_3 + 3 . C_4 . 8 = 35$

$$\left[\begin{array}{cccc|c} 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 2 & 3 \\ 4 & 1 & 2 & 8 & 12 \\ 8 & 1 & 3 & 24 & 35 \end{array}\right] \Rightarrow$$

$C_1 = -2$

$C_2 = -$

$t_n = -2 - n + 2^{n+1} + n.2^n$

$t_n \in O(n.2^n)$

Bubble Sort: "Psedo Code" (5. hafta)

"Yalancı kod"

```
# Bubble sort (A)
for i <- 1 to Lenght [A]
    do for j <- Lenght[A] downto i+1
        do if A[j] < A[j-1]
            then exchange A[j] <-> A[j-1]
```

Ex:

	8	20	12	4	5	18	2
i=1	8	20	12	4	5	2	18
j=7=2	8	20	12	4	2	5	18
j=6=2	8	20	12	2	4	5	18
j=5=2	8	20	2	12	4	5	18
j=4=2	8	2	20	12	4	5	18
j=3=2	2	8	20	12	4	5	18
j=2=2							

| i=2

$$\sum_{i=1}^{n}\sum_{j=2}^{n} 1$$

$(n-2+1) \cdot 1$

Bu şekilde sıralama algoritması oldu. "verimli değil" "karmaşıklığa sahip"

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

insurtion sort: Sırasına göre uygun yere.

| Sorted | unsorted | ← sıralanmamış kısım.

"1. eleman sıralanmış düşünüyoruz"

```
# Insurtion _sort (A)
for j <- 2 to Lenght [A]
    do key <- A[j]
    ▷ insert A[j] in to the sorted sequence A[1..j-1]
    i <- j-1
    while i>0 And key < A[i]
        do A[i+1] <- A[i]
           i <- i-1
    A[i+1] <- key
```

Ex:

"sıralanmamış dizi uygun yere koymak"

Ex.:

1	2	3	4	5	6
23	78	45	8	32	6

sorted   unsorted.

→

1	2	3	4	5	6
23	78	45	8	32	6

sorted   unsorted

$j=2$, key $\leftarrow$ 78 , $i = j-1$ // i azaldı //

$j=3$: key $\leftarrow$ 45 $i=2$ →

23	45	78	8	32	6

sorted   unsorted.

$j=4$, key $\leftarrow$ 8. $i=3$ →

23 45 8 78 32 6

23 8 48 78 32 6

8 23 48 78 | 32 6

sorted   unsorted.

$$\sum_{j=2}^{n} \sum_{i=0}^{j-1} 1 \Rightarrow \sum_{i=0}^{j-1} \underbrace{(j-1+1)}_{j} . 1$$

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

Worst case $\Rightarrow O(n^2)$ kötü

Best case $\rightarrow O(n)$

// Bubble'den daha iyi //

Selection sort: // her defasında en küçüğünü seçip alıyoruz //

```
for i ← 1 to n-1
    min ← i
        for j ← i+1 to n
            if A[j] < A[min] then
                min ← j
        if min ≠ i then exchange A[i] ⇔ A[min]
```

رح ياخذ الاصغر

عدد تي

// الرينزن

Ex.: sorted | 5   2   4 , 6   1   3

unsorted.

1 | 2   4   5   3

sorted

$i=1$ $min=1$ $j=2$

$\cancel{2}$ $j = \cancel{3}, \cancel{4}, 5$

$min=5$ $j=6$

$A[i] \leftrightarrow A[5]$

A[3] < A[4]

A[j]

CamScanner ile tarandı

1 2 | 4 6 5 3

Sorted

$i=2$, $min=2$, $j=3$

~~4~~, ~~5~~, ~~6~~

$i=3$, $min=3$, $j=~~4~~$

~~5~~, 6

1 2 3 | 6 5 4

Sorted | unsorted.

$A[3] \Leftrightarrow A[6]$

$i=4$, $min=4$, $j=5$

6

$A[4] \Leftrightarrow A[6]$

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

3- Selection Sort :

1- for $i \leftarrow 1$ to $n-1$
  $min \leftarrow i$
  for $j \leftarrow i+1$ to n
    if $A[j] < A[min]$ then
      $min \leftarrow j$
  if $min \neq i$ then exchange $A[i] \leftrightarrow A[min]$

ör/

sorted | 5 2 4 6 1 3
unsorted

1 en küçük eleman derek

$i=1$, $min = \cancel{1}$ $\cancel{2}$ 5 $J: \cancel{2}$ $\cancel{3}$ $\cancel{4}$ 5

$A[1] \leftrightarrow A[5]$

1 | 2 4 6 5 3
sorted | unsorted

$i=2$ $min=2$ $J=3$

1 2 | 4 6 5 3
sorted | unsorted

$i=3$ $min=\cancel{3}$ 6 $J=4$

$A[3] \leftrightarrow A[6]$

1 2 3 | 6 5 4
sorted | unsorted

Karmaşıklığı $= (n-1) + (n-2) + \ldots + 1 = \frac{(n-1).(n)}{2} = \theta(n^2)$

Böl ve Yönet =>

- Merge sort
- Quick Sort
- Heap sort

1- Merge Sort (A,p,r)

if $p < r$
then $q \leftarrow \lfloor (p+r)/2 \rfloor$
  MERGE - SORT (A,p,q) → rekürsif fonk
  MERGE - SORT (A, q+1, r)
  MERGE (A,p,q,r)

MERGE (A,p,q,r)

$n_1 \leftarrow q-p+1$
$n_2 \leftarrow r-q$
Create

11

Date : ....../....../......

MERGE (A, p, q, r)

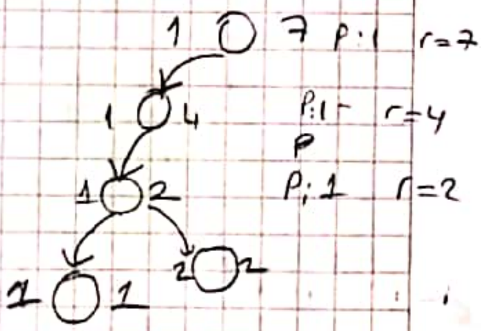
$n_1 \leftarrow q-p+1$
$n_2 \leftarrow r-q$

create arrays $L[1 \ldots n_1+1]$ and $R[1 \ldots n_2+1]$
for $i \leftarrow 1$ to n_1
    do $L[i] \leftarrow A[p+i-1]$
for $j \leftarrow 1$ to n_2
    do $R[j] \leftarrow A[q+j]$
$L[n_1+1] \leftarrow \infty$
$R[n_2+1] \leftarrow \infty$
$i \leftarrow 1$
$j \leftarrow 1$
for $k \leftarrow p$ to r
    do if $L[i] \leq R[j]$
        then $A[k] \leftarrow L[i]$
            $i \leftarrow i+1$
        else
            $A[k] \leftarrow R[j]$
            $j \leftarrow j+1$

Ör: A: | 38 | 27 | 43 | 3 | 9 | 82 | 10 |

$p=1$ $r:7$

$(p+r/2)\ q=4$



MERGE-SORT $(A,1,4)$

MERGE-SORT $(A,1,2)$

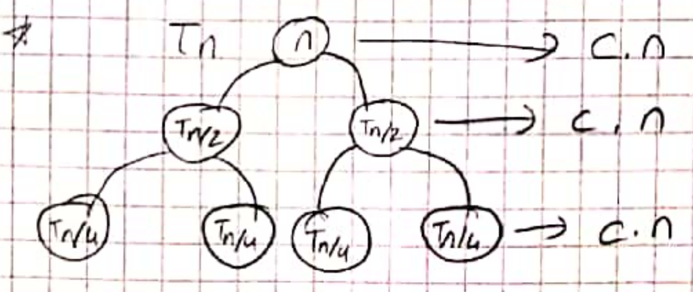
MERGE-SORT $(A,1,1)$

MERGE-Sort $(A,2,2)$

MERGE $(A, \overset{p}{1}, \overset{q}{1}, \overset{r}{2})$

n_1 $q-p+1 = 1$ L | 38 | ∞ |
n_2 $r-q = 1$ R | 27 | ∞ |
$i=1$
$j=1$

for $k \in 1$ to 2



$\log n \cdot c.n \Rightarrow c.n.\log n$

$\Theta(n.\log n)$

CamScanner ile tarandı

9 12 Algoritma Analizi ve Tasarımı :

Subject : Date : ....../....../......

Quick Sort :

Pivot : P

≤	P	>

in-place :

P.Code =) ~~Quick~~

```
Quick Sort (A,p,r)
   if p<r
      then q ← PARTITION (A,p,r)
           QuickSort (A,p,q-1)
           QuickSort (A,q+1,r)
```

PARTITION (A,p,r) $\boxed{n \log n}$ karmaşıklığı

```
Pivot → x ← A[r]
        i ← p-1
        for j ← p to r-1
            do if A[j] ≤ x
                 then i ← i+1
                      exchange A[i] ↔ A[j]
        exchange A[i+1] ↔ A[r]
dir     return i+1
```

85 24 63 45 12 31 96 50

q ← portition (A, 1, 8)

x ← 50 J = 2 4

i = 0+1 = 0 → 1+1 = 2 → 2+1 = 3

24	85	63	45	12	31	96	50
24	45	63	85	12	31	96	50
24	45	12	85	63	31	96	50
24	45	12	31	63	85	96	50
24	45	12	31	$\boxed{50}$	85	96	63

≤ >