ALGORİTMALAR

19.02.19

Hafta - 2

Insertion Sort

```
for i=1 to n-1
    pivot = A[i]
    j = i-1
    while j>=0 and A[j]>pivot
        A[j+1] = A[j]
        j = j-1
    A[j+1] = pivot
```

i = 0 → 3
1 → 7
2 → 12
3 → 4
4 → 6
5 → 2
6 → 1
7 → 5

pivot = 7
i = 1 j = 0
pivot = 12
i = 2 j = 1
pivot = 4
i = 3 j = 2
A[3]'e A[2]'yi yazdık
⋮

$$\sum_{i=1}^{n-1} i \Rightarrow \underbrace{f(n) = \sum_{i=1}^{n-1} \sum_{j=i-1}^{0} 1}_{\text{En Kötü Durum}}$$

$$\underbrace{\sum_{i=1}^{n-1} 1}_{\text{En İyi Durum}} \Rightarrow \text{while'a bir kez baktık (sıralıydı)}$$

$$\sum_{i=1}^{n-1} i = 1+2+3+4+\dots+n-1$$
$$= \frac{n.(n-1)}{2} \Rightarrow O(n^2) \text{ en kötü durum}$$

$$\sum_{i=1}^{n-1} 1 = n-1 \Rightarrow O(n) \text{ en iyi durum}$$

Soru: 5 çift çorabı yıkıyoruz. Bu çoraplardan 2 tanesi kayboluyor. Kaç çift kalır?

En iyi durum → 4
En kötü durum → 3
Beklenen → $\frac{28}{9}$

$\underset{\text{Tüm Olay}}{\binom{10}{2}}$ $\underset{\text{En iyi}}{\binom{5}{1}} \Rightarrow \frac{5}{45} = \frac{1}{9}$ En iyinin olasılığı

$\frac{8}{9}$ En kötünün olasılığı

$\frac{1}{9} \cdot 4 + \frac{8}{9} \cdot 3 = \frac{28}{9}$ → Beklenen

Polinomal Değerlendirme

$P(x) = a_i x^i + \dots + a_3 x^3 + a_2 x^2 + a x + a_0$

```
top=0
for j=0 to i
    top = top + a[j] * pow(x,j)
```

iyileştirme →

$$\underbrace{\sum_{j=0}^{i} j}_{\text{çarpma işlemi}} \Rightarrow 0+1+2+\dots+i$$

$$\frac{i(i+1)}{2}$$

NOT! Bir önceki üssü elimizde tutarsak algoritma iyileşir.

```
top=0
int önceki=1
for j=0 to i
    top = top + a[j] * önceki
    önceki = önceki * x
```

i+1

oo Asimptotik Büyüme Dereceleri oo

- $O(g(n)) \rightarrow$ (Big Oh) Ya $g(n)$'le aynı hızda büyüyorlar ya da $g(n)$'den daha düşük hızla büyüyorlar.
- $\Theta(g(n)) \rightarrow$ (Big Theta) $g(n)$'le aynı oranda büyüyorlar.
- $\Omega(g(n)) \rightarrow$ (Big Omega) Ya $g(n)$'e aynı hızda büyüyorlar, ya da $g(n)$'den daha yüksek hızla büyüyorlar.

o Soru: $\frac{1}{2}n(n+1) \in O(g(n))$ inceleyiniz

$$\frac{n^2+n}{2} \rightarrow \left(\frac{n^2}{2}+\frac{n}{2}\right) \leq c.g(n), \quad \forall n \geq n_0$$

NOT! $g(n)$ yerine tahmini bir fonksiyon veriyoruz. c yerine de herhangi bir sabit sayı verip sağlayıp sağlamadığına bakıyoruz.

$$\left.\begin{aligned} g(n) &= n^3 \\ n_0 &= 1 \\ n_0 &= 2 \end{aligned}\right\} \quad \frac{n^2}{2}+\frac{n}{2} \leq 1.n^3 \checkmark \quad \frac{n^2}{2}+\frac{n}{2} \leq 2.n^3 \checkmark$$

NOT! Evde $g(n)$ yerine n^2 koyarak dene!!!

$$\left.\begin{aligned} g(n) &= n^2 \\ n_0 &= 1 \\ n_0 &= 2 \end{aligned}\right\} \quad \frac{n^2}{2}+\frac{n}{2} \leq 1.n^2 \checkmark \quad \frac{n^2}{2}+\frac{n}{2} \leq 2.n^2 \checkmark$$

Algoritmalar

26.02.19

Hafta-3

Big O: Yukarıdan sınırlandırır

$$f(n) \le c\,g(n), \quad \forall n \ge n_0$$

Örn: $n(n-1) \in O(n^3)$

$n^2 - n \le n^3 \cdot c$

$c=1 \Rightarrow n^2 - n \le n^3 \cdot 1 \quad n_0 \ge ?$

$n_0 \ge 1$

Örn: $3^{n+1} + 3^{n-1} \in O(?)$

$3^n \cdot 3 + 3^n \cdot \frac{1}{3}$

$3^n \underbrace{\left(3 + \frac{1}{3}\right)}_{\text{en yakın } 4} \le 3^n \cdot 4$

$c=4 \quad n_0 \ge 0$

Örn: $f(n) = 3n^2 + 4n + 1 \in O(?)$

$3n^2 + 4n + 1 \le c \cdot n^2$

$3n^2 + 4n + 1 \le 8n^2$

$4n + 1 \le 5n^2 \quad n_0 \ge 1$

$f(n) \in O(n^2)$

Örn: $n \in O(n^2)$ ✓

Örn: $100n + 5 \in O(n^2)$ ✓

$100n + 5 \le 100n + n \le 101n^2$

$100n + 5 \le 101n \quad n_0 \ge 5$

$\underline{100n + 5 \in O(n^2)}$

Özellikler:

- $f(n) \in O(f(n))$
- $f(n) \in O(g(n))$ ise $g(n) \in \Omega(f(n))$
- Eğer $f(n) \in O(g(n))$ ve $g(n) \in O(h(n))$ ise $f(n) \in O(h(n))$
- $f_1(n) \in O(g_1(n))$ $f_2(n) \in O(g_2(n))$;
  $f_1(n) + f_2(n) \in O(\max\{g_1(n), g_2(n)\})$

İspat: $f_1(n) \le c_1 \cdot g_1(n) \quad n_0 \ge n_{01}$

$+ \quad f_2(n) \le c_2 \cdot g_2(n) \quad n_0 \ge n_{02}$

$f_1(n) + f_2(n) \le \underline{c_1} \cdot g_1(n) + \underline{c_2}\, g_2(n)$ — c_1 ve c_2'den büyük olan

$f_1(n) + f_2(n) \le c_1 g_1(n) + c_2 g_2(n) \le 2c\, O(\underbrace{\max(g_1(n), g_2(n))}_{g_1 \text{ ve } g_2\text{'den büyük olan}})$

Big Ω: Aşağıdan sınırlandırır

$$f(n) \ge c\,g(n), \quad \forall n \ge n_0$$

Örn: $0{,}05n^{10} + 3n^3 + 1 \in \Omega(n^{10})$

$0{,}05n^{10} + 3n^3 + 1 \ge n^{10} \cdot c$

$c = 0{,}05$

$n \ge 1$

Örn: $n^3 \in \Omega(n^2)$

$n^3 \ge c \cdot n^2$

$n^3 \ge n^2$

$n \ge 1$

Örn: $n(n-1) \in \Omega(n)$

$n^2 - n \ge n \cdot c \quad c = 1$

$n^2 - n \ge n$

$n^2 \ge 2n$

$n \ge 2$

$n(n-1) \in \Omega(n)$

Big Θ: Büyüme derecesi aynıdır. Aşağıdan ve yukarıdan sınırlar.

$$c_1 g(n) \le f(n) \le c_2 g(n), \ \forall n \geq n_0$$

Örn! $\frac{1}{2}n(n-1) \in \Theta(n^2)$

$$\frac{n^2}{2} - \frac{n}{2} \le n^2 \cdot c \qquad \frac{n^2}{2} - \frac{n}{2} \geq n^2 \cdot c$$

$C_2 = \frac{1}{2}$ $\quad n_{01} \geq 1$ $\quad c_1 = \frac{1}{4}$ $\quad -\frac{1}{2}n \geq -\frac{1}{4}n^2$

$\frac{1}{2}n \le \frac{1}{4}n^2$

$n_0 = \max(n_{01}, n_{02})$ $\quad 2 \le n_{02}$

$= 2$

$\frac{1}{2}n(n-1) \in \Theta(n^2)$ ✓

$n_0 = 2$

Limit Kullanarak Belirleme

$$\lim_{n\to\infty} \frac{T(n)}{G(n)} \begin{cases} * \ 0 \text{ çıkarsa } T(n) \in O(G(n)) \\ * \text{ sabit çıkarsa } T(n) \in \Theta(G(n)) \\ * \ \infty \text{ çıkarsa } T(n) \in \Omega(G(n)) \end{cases}$$

NOT! $n! \approx (2\pi n)^{1/2} (n/e)^n$

Örn! $\underbrace{\frac{1}{2}n(n-1)}_{f(n)}$, $\underbrace{n^2}_{g(n)}$ büyüme hızlarını limitle kıyasla

$$\lim_{n\to\infty} \frac{\frac{1}{2}n^2 - \frac{1}{2}n}{n^2} = \lim_{n\to\infty} \left(\frac{1}{2} - \underbrace{\frac{1}{2n}}_{0}\right) = \frac{1}{2}$$

$\boxed{\frac{1}{2}n(n-1) \in \Theta(n^2)}$

Örn! $\log_2 n$, $\sqrt{n}$?

$$\lim_{n\to\infty} \frac{(\log_2 n)'}{(\sqrt{n})'} = \frac{\log_2 e \cdot \frac{1}{n}}{\frac{1}{2\sqrt{n}}} = \log_2 e \cdot \lim_{n\to\infty} \frac{1}{n} \cdot 2\sqrt{n} = 0$$

NOT! $\log_e n' = \frac{1}{n}$
$\log_2 n = \log_2 e \cdot \log_e n$

$\boxed{\log_2 n \in O(\sqrt{n})}$

$\boxed{\sqrt{n} \in \Omega(\log_2 n)}$

Örn! $n!$, 2^n = ?

$$\lim_{n\to\infty} \frac{n!}{2^n} = \frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n}{2^n} = \sqrt{2\pi} \lim_{n\to\infty} \frac{\sqrt{n}\left(\frac{n}{e}\right)^n}{2^n} = \sqrt{2\pi} \lim_{n\to\infty} \frac{\sqrt{n} \cdot n^n}{2^n \cdot e^n} = \infty$$

$\boxed{n! \in \Omega\ 2^n}$

$\boxed{2^n \in \Theta\ n!}$

Örn! $\log_a n \quad \log_b n$ $\qquad$ * $a^n \notin O(b^n)$

$\log_a n = \underbrace{\log_a b}_{c} \log_b n$

???

Recursive Olmayan Algoritmaların Zaman Analizi

1. Girdi boyutu için parametreye karar verme
2. Algoritmada en çok çalışan işlemi bulma (while, for)
3. En iyi, en kötü durumu bulma
4. Toplam formülü belirleme
5. Sadeleştirme

Algoritma 1: n elemanlı arrayin max elemanı

```
for i = 1 to n-1 do
    if A[i] > maxval
        maxval = A[i]
return maxval
```

Temel işlem: if →

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

Algoritma 2: n elemanlı arrayin tüm elemanları birbirinden farklı mı?

```
for i = 0 to n-2
    for j = i+1 to n-1
        if A[i] = A[j]
            return false
return true
```

Temel işlem: if →

$$\sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1$$

$$\sum_{i=0}^{n-2} ((n-1)-i) \Rightarrow \sum_{i=0}^{n-2} (n-1) - \sum_{i=0}^{n-2} i$$

$(n-1)-(i+1)+1$ o kadar dönüyor

$n-2-0+1$

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

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

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

Algoritma 3 : Matris çarpımları

```
for i = 0 to n-1
    for J = 0 to n-1
        C[i,J]
        for k = 0 to n-1
            C[i,J] + A[i,k] * B[k,J]
return C
```

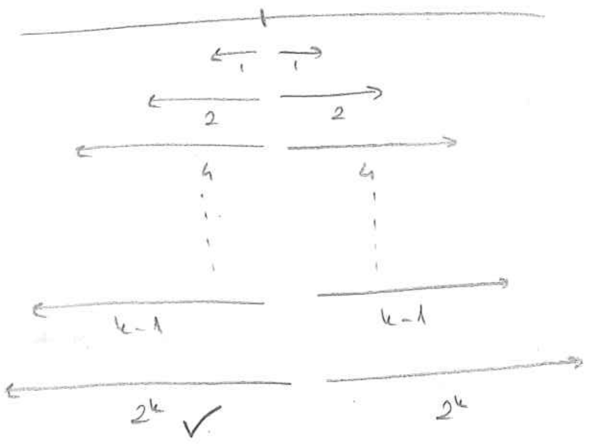
$$M(n) = \sum_{i=0}^{n-1} \sum_{J=0}^{n-1} \sum_{k=0}^{n-1} 1 = \sum_{i=0}^{n-1} n^2 = n^3$$

Soru: n bilye var. Biri sahte sadece eşit kollu terazi var. Sahte olan diğerlerinden daha mı hafif daha mı ağır?

Algoritmalar

05.03.19

Hafta-4



ilk adımda 1 gidiyorum

$$\underbrace{4\sum_{i=1}^{k-1} 2^{i-1}} + 3.2^k$$

$$4(2^k-1)+3.2^k = 4.2^k+3.2^k-4$$

$$\Rightarrow 7.2^k-4 \le c_1.k \le c_2.2^k$$

Soru: n elemanlı n. dereceden n tane denklemimiz var.

$\frac{1}{A_{00}}$ $-A_{10}$

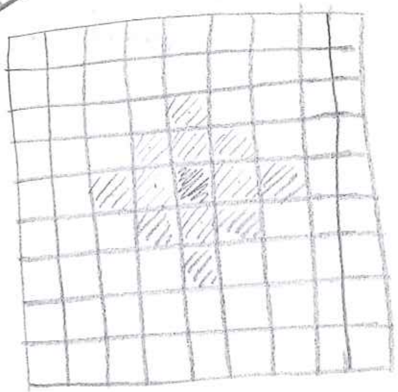
$$\begin{bmatrix} A_{00}x_1 & A_{01}x_2 & A_{02}x_3 & \cdots & A_{0n-1}x_n \\ A_{10}x_1 & \cdots & & & \\ \vdots & & & & \\ A_{n-10}x_n & \cdots & & & A_{n-1n-1}x_n \end{bmatrix} \begin{matrix} = C_0 \\ = C_1 \\ \\ = C_{n-1} \end{matrix}$$

```
for i=0 to n-2
    for j=i+1 to n-1
        for z=i to n
            A[j,z] ← (-1/A[i,i] A[j,i]) x [A[i,z]] + A[j,z]
```

$$\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1}\sum_{z=i}^{n}$$

Haftaya karmaşıklığı yaz

Soru



0	1	2	3
1,	5	13	25

$\underbrace{1 \quad 5}_{4} \quad \underbrace{5 \quad 13}_{8} \quad \underbrace{13 \quad 25}_{12}$

$$4\sum_{i=1}^{n} i + 1 = \frac{4n(n+1)}{2} + 1 = 2n(n+1)+1$$
$$= 2n^2 + 2n + 1$$

Özyineleme İlişkileri

- İleriye doğru yerine koyma
- Geriye doğru yerine koyma
- Diferansiyel denklemler

* Master teoremi (Yaklaşık bir sonuç veriyor)

Örn: Faktöriyel Hesaplaması

```
if n=0 return 1
else return F(n-1)*n
```

$M(n) = M(n-1) + 1$ → toplam kaç kere çarpma işlemi gerçekleştiriyor

for $n > 0$

$\underbrace{M(0) = 0}_{\text{Forward}}$

$M(0) = 0$

$M(1) = \underbrace{M(0)}_{0} + 1$

$M(2) = M(1) + 1$

$M(3) = M(2) + 1$

$M(3) = 1 + 1 + M(0) + 1$

$M(4) = M(3) + 1$

$= M(2) + 1 + 1$

$= M(1) + 1 + 1 + 1$

$= M(0) + 1 + 1 + 1 + 1$

$\boxed{M(n) = n \in \Theta(n)}$

$M(n) = M(n-1) + 1$

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

$M(n) = M(n-3) + 1 + 1 + 1$

$M(n) = M(n-4) + 1 + 1 + 1 + 1$

$\underbrace{M(n) = M(n-i) + i}_{\text{Backward}}$

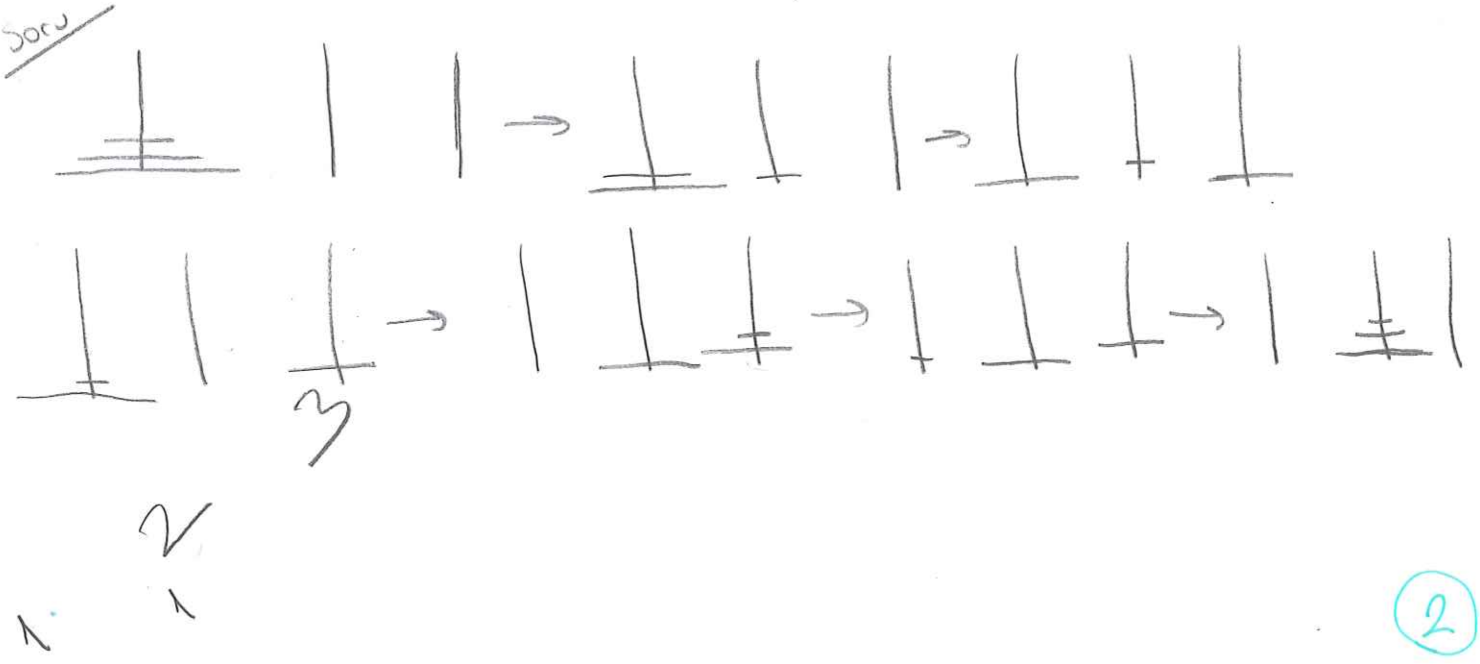
$n - i = 0$

$i = n$

$M(n) = \underbrace{M(0)}_{0} + n$

$M(n) = n \in \Theta(n)$

Soru



$M(n) = 2 * M(n-1) + 1$

$M(1) = 1$

$M(n) = 2[2 * M(n-2) + 1] + 1$

$M(n) = 2[2[2 * M(n-3)+1]+1]+1 = 2^2 M(n-2) + 2 + 1$

$M(n) = 2[2[2 * M(n-i)+1]+1]+1 = 2^3 M(n-3) + 4 + 2 + 1$

$- - - - - - - - - - - - = 2^4 M(n-4) + 8 + 4 + 2 + 1$

$M(n) = 2^i M(n-i) + 2^{i-1} + 2^{i-2} + \dots + 2^0$

$= 2^i M(n-i) + 2^i - 1 \qquad M(1) = 1 \qquad 1 = n-i \quad i = n-1$

$\underline{2^{n-1} M(1) + 2^{n-1} - 1}$

~~Soru~~

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

$T(2) = 0$

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

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

$= 2^2 (2(T(n^{1/2^3}) + 1)) + 2 + 1$

$= 2^3 T.(n^{1/2^3}) + 4 + 2 + 1$

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

$\log_n(n^{1/2^i}) = (2) \log_n$

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

$\frac{1}{\log_n 2} = 2^i \Rightarrow \log_2 n = 2^i$

$\log_2 \log_2 n = i$

$\underline{2^{\log \log_2 n} 0 + 2^{\log_2 \log_2 n} - 1}$

$\in \Theta(\log n)$

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

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

$\log_2 n = 2^i$

$\log_2 \log_2 n = i$

3

Algoritmalar

12.03.19

Hafta-5

$$X(n) = X(n-2)+2 \quad , \text{ for } n>1 \quad X(1)=0$$
$$X(n-2) = X(n-4)+2$$
$$X(n-4) = X(n-6)+2$$
$$X(n) = X(n-6)+2+2+2$$
$$X(n) = X(n-i)+i \qquad n-i=1$$
$$i=n-1$$
$$X(n) = X(n-n+1)+n-1$$
$$X(n) = \underbrace{X(1)}_{0}+n-1 \in \Theta(n)$$

$$X(n) = 2X(n-3) \quad , \text{ for } n \geq 1 \quad X(1)=1$$
$$X(n-3) = 2X(n-6)$$
$$X(n) = 2.2X(n-6)$$
$$X(n-6) = 2X(n-9)$$
$$X(n) = 2.2.2X(n-9)$$
$$X(n) = 2^i X(n-3i) \qquad n-3i=1$$
$$i = \frac{n-1}{3}$$
$$X(n) = 2^{\frac{n-1}{3}} . X(1) \in \Theta(2^{n/3})$$

$$X(n) = X(n/5)+n \quad \text{for } n \geq 1 \quad X(1)=1$$
$$n = 5^k$$
$$X(5^k) = X(5^{k-1})+5^k$$
$$X(5^{k-1}) = X(5^{k-2})+5^{k-1}$$
$$X(5^{k-2}) = X(5^{k-3})+5^{k-2}$$
$$X(5^k) = X(5^{k-3})+5^{k-2}+5^{k-1}+5^k + \dots 5^1$$
$$X(5^k) = X(5^{k-i})+5^{k-i+1}+5^{k-i+2}+\dots+5^1$$
$$i=k$$
$$X(5^i) = \underbrace{X(5^0)}_{1}+5^1+5^2+\dots$$
$$\frac{5^{k+1}-1}{5-1} \Rightarrow \frac{5n-1}{4} \in \Theta(n)$$

Fibonacci Numbers

$$F(n) = \frac{1}{\sqrt{5}}(\phi^n - \hat{\phi}^n) \quad \phi = (1+\sqrt{5})/2$$

$$A(n) = A(n-1)+A(n-2)+1.$$ → fibonacci sayısında kaç tane toplama işlemi yapılır?

Brute Force *

Sort *

Selection Sort

```
z = 0
for i=0 to n-2
    min ← A[i]
    for J = i+1 to n-1
        if min > A[J]
            z ← J
            min = A[J]
    swap A[i], A[z]
```

$$\frac{n(n-1)}{2} \in \Theta(n^2)$$

Bubble Sort

```
for i=0 to n-1
    for J=0 to n-1-i
        if A[J+1] < A[J]
    swap A[J], A[J+1]
```

$\frac{n(n-1)}{2} \in \Theta(n^2)$

String Matching

T → OBJECTORIENTEDCLASS → n tane

P → [CLASS] → m tane

```
for i=0 to n-m
    J ← 0
    while T[i+J] = P[J]
        J ← J+1
    if (J == m)
        return i
return -1;
```

$\Theta((n-m+1)(m))$

???

```
for i=0 to n-2
    if(P[i] = "A")
        for(J=i; J>i; J++) {
            P(J) = "C"
            sayi++;
        }
```

n^2

CAABCACBAC

A sayisi +

top = top + A sayisi

⇓

n

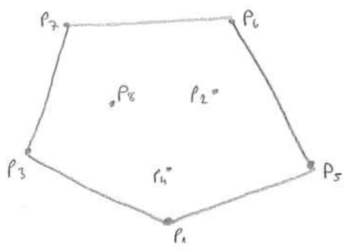
Algoritmalar

19.03.19
Hafta-6

Nokta işlemi → $\binom{n}{2} = \frac{n(n-1)}{2}$ (c1) $\in \Theta(n^2)$

Convex-Hull Problemi

Elimizde çeşitli noktalar var bu noktalar kümesinin etrafına çizebileceğimiz minimum dışbükey



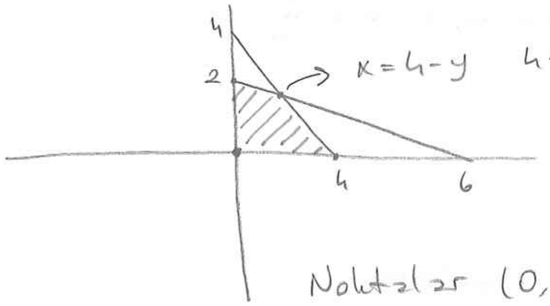
Noktalar arasındaki doğru denklemlerine bakarız eğer bulduğumuz denklem diğerlerini tek bir tarafta bırakıyorsa sınır olduğunu anlarız.

$\frac{n(n-1)}{2} \cdot (n-2) \in \Theta(n^3)$

Amaç maximize $3x+5y$

Kısıtlar $x+y \leq 4$
$x+3y \leq 6$
$x \geq 0,\ y \geq 0$

Simplex Yöntemi



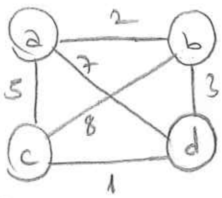
$x = 4-y$

$4-y+3y=6$
$4+2y=6$
$y=1 \quad x=3$

Noktalar (0,2) (1,3) (4,0) (0,0)
↓ ↓ ↓ ↓
10 14 12 0

Exhaustive Search (Kapsamlı Arama)

** Gezgin Satıcı Problemi **



$a \to \boxed{b \to d \to c} \to a = 11$
$a \to \boxed{b \to c \to d} \to a = 18$
$a \to c \to d \to b \to a = 11$
$a \to c \to b \to d \to a = 23$
$a \to d \to b \to c \to a = 23$
$a \to d \to c \to b \to a = 18$

Kaç Yol Var?
$(n-1)!$

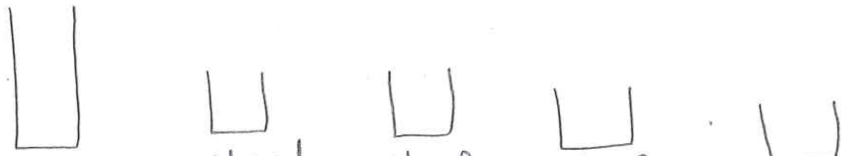
$\frac{(n-1)!}{2} \in \Theta(n!)$

aynı yol kümeleri var

NOT! N-P Hard
Non-deterministicly polynominal

1

Knapsack Problemi



kapasite = 10

	item 1	item 2	item 3	item 4
	$V_1 = 42$ TL	$V_2 = 12$ TL	$V_3 = 40$ TL	$V_4 = 25$ TL
	$w_1 = 7l$	$w_2 = 3l$	$w_3 = 4l$	$w_4 = 5l$

$2^4 \to$ kombinasyon

{1} {1,2} .
{2} {1,3} .
{3} .
{4} . {1,2,3,4}

NOT ! Kombinasyon permütasyoları kodla nasıl oluştururum??

Assignment (Atama) Problemi

	Job 1	2	3	4
Person 1	9	2	7	8
2	6	4	3	7
3	5	8	1	8
4	7	6	9	4

birinci kişi maaliyeti

$9+4+1+4=18$

$2+6+1+4=13$

$$P \to \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \hline 4 & 3 & 2 & 1 \\ 4 & 2 & 3 & 1 \end{array} \Big\} \, n!$$

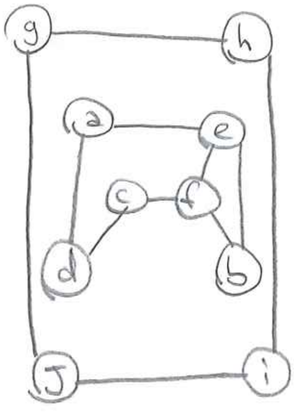
NOT ! Hungarian algoritması ile daha kolay çözülebiliyor

ÖDEV PROBLEM

$$\begin{array}{r} \text{SEND} \\ +\ \text{MORE} \\ \hline \text{MONEY} \end{array}$$

* Her bir harf bir rakama denk geliyor bu sayıların ne olduğunu bul.

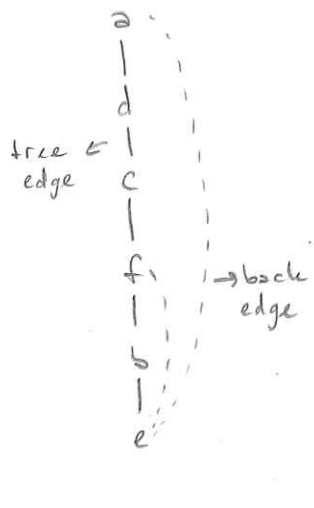
Graph Traversal Algoritmaları ⇒ DFS BFS



a, d, c, f, b, e

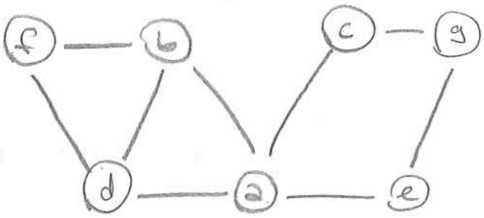
dfs(a)	dfs(g)
$e_{6,1}$	$J_{10,7}$
$b_{5,2}$	$i_{9,8}$
$f_{4,3}$	$h_{8,9}$
$c_{3,4}$	$g_{7,10}$
$d_{2,5}$	
$a_{1,6}$	

unconn.



2

DFS



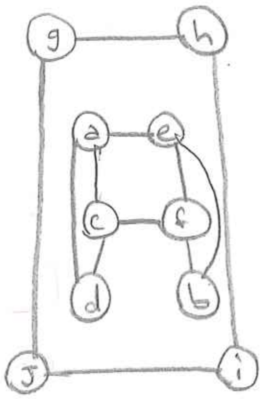
dfs(a)

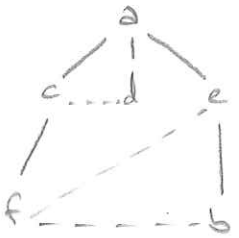
e 7,1
g 6,2
c 5,3
f 4,4
d 3,5
b 2,6
a 1,7

$|V|+|E|$

Cormen - Algoritmalar

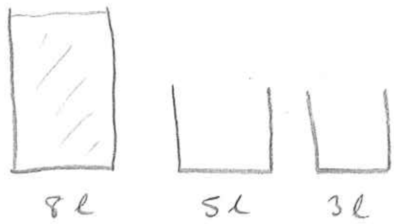
BFS



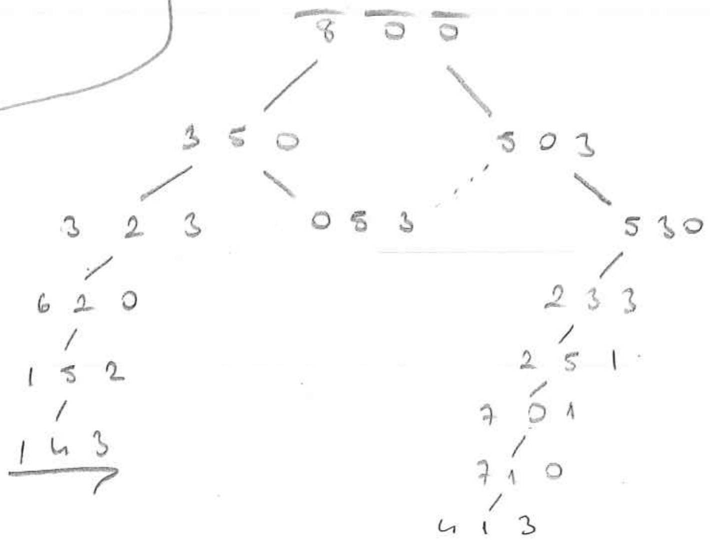


Sınavda DFS, BFS'le ilgili sorular çıkabilir

Soru:



Bir kavanozda 4l'yi nasıl oluştururuz?



26.03.19

Hafta-7

Algoritmalar

Decrease And Conquer

* Parçalardan bir tanesini seçiyor onu çözüyoruz.
  - Top-Down Yaklaşım: Büyük parçadan ufağa inme (Recursive)
  - Bottom-Up Yaklaşım: Ufak parçaları keşfedip yukarıya çıkma

** $a^n = a^{n-1} \cdot a \qquad f(n) = a^n$

Top-Down

$$f(n) = \begin{cases} f(n-1) \cdot a & \text{if } n > 1 \\ a & \text{if } n = 1 \end{cases}$$

Bottom-Up

$(n-1)$ defa çarpım yapılabilir

Örn: a^n i hesapla

$a^n = (a^{n/2})^2 \longrightarrow$ problemi ikiye böldük

tek bir parçayı çözmeye çalışıyoruz (Decrease and Conquer)

Decrease by constant

* Insertion Sort
* Topological Sort
* Permütasyon Kombinasyon

Decrease by a constant factor

* Binary search ve bisection
* Üs alma
* Rus çarpımı

Sorting

n elemanlı arrayi sıralama (Insertion Sort) (Bottom-Up)

```
for i = 1 to n-1
    temp = A[i]
    J = i - 1
    while J >= 0 and A[J] > temp
        A[J+1] = A[J]
        J = J - 1
    A[J+1] = temp
```

$$\sum_{i=1}^{n-1} \sum_{J=i-1}^{J=0} 1$$

$$\sum_{i=1}^{n-1} i \quad \frac{(n-1)(n)}{2} \in \Theta(n^2)$$

* ShellSort iyileştirilmiş hali

Permütasyon

* 1'den n'e kadar sayıların permütasyonu
"Decrease by one" → (n−1) eleman için permütasyonları oluştururuz

1

12 21

3 3

123 132 312 | 321 231 213 → minimal change

1234 1243 1423 4123 | 4132 1432 1342 1324 . . .

soldan ekleme

Algorithm Johnson Trotter

←←← 123 ←←← 132 ←←← 312 →←← 321 ←→← 231 ←←→ 213

* En büyük 3 o yüzden değiştik
* Oklar değişmedi çünkü 3 büyük
* 3'ün dev kendisinden küçüğü göstermiyor. Bu yüzden 2 mobil oldu
* 2 mobil olduğu için mobil olmayan en büyüğün okunun yönü değişti.

←←←← 1234 ←←←← 1243 ←←←← 1423 ←←←← 4123 →←←← 4132 ←→←← 1432

* mobil 4
* mobil 3
* mobil 4

Algorithm Lexicographic Permute

362541

* Sağda azalan kümeye bakıyoruz. Bu kümeyi sağlamayan eleman kim? 2
* Azalan kümede 2'den büyük en küçük sayı nedir? 4
* 2'nin yerine 4'ü yazıyor. Geriye kalanı küçükten büyüğe sıralıyoruz.

364|125

364152 1

364215

364251

3

Algoritmalar

02.04.19

Hafta-8

Binary Reflected Gray Code

$L=\{0,1\} \leftarrow BRGC(1)$ BRGC(3)

BRGC(2) →

$L_1=\{00,01\}$ $L_1=\{000,001,011,010\}$

$L_2=\{11,10\}$ $L_2=\{110,111,101,100\}$

$L=\{000,001,011,110,111,101,100\}$

Algorithm : HeapPermute

```
if n=1
   write A
else
   for i=1 to n
      HeapPermute(n-1)
      if n is odd
         swap A[1] and A[n]
      else swap A[i] and A[n]
```

A=[1 2]

Heappermute(2)   n=2

1 2

A=[2 1]

2 1

A=[1 2 3]

HeapPermute(3)   n=3

i=1 → heappermute(2)

↓

i=1

heappermute(1)

1 2 3

else girdik swap ettik

Devam → ~~if n is odd~~

swap

A=[3 1 2]

i=2 → heappermute(1)

2 1 3

※ HeapPermute(4)'ü çalıştırıp dene (A=[1 2 3 4])

1 2 3

A=[2 1 3]

2 1 3

A=[3 1 2]

3 1 2

A=[1 3 2]

1 3 2

A=[2 3 1]

2 3 1

A=[3 2 1]

3 2 1

Sınav Soruları

2018 1-) a-) $n(n+1) \notin O(n^4)$ YANLIŞ

$n^2+n \leq n^4 \cdot c$

$c=1 \quad n \geq 2$

b-) $2^{n-1}+n \in \Omega(2^n)$ DOĞRU

$c \cdot 2^n \leq 2^{n-1}+n$

$c=\frac{1}{4} \quad 2^{n-2} \leq 2^{n-1}+n$

$n \geq 2$

c-) $n(n+1) \in \Theta(n^3)$ YANLIŞ

d-) $9n^4-7n^3+3 \in \Theta(n^4)$ DOĞRU

$$\lim_{n\to\infty} \frac{9n^4-7n^3+3}{n^4} = 9 - \frac{7}{n} + \frac{3}{n^4} = 9$$

e-) $\Theta(\alpha g(n)) \in \Theta(g(n)), \alpha \in N^+$ DOĞRU

$$\underbrace{\frac{\alpha}{2}}_{c_1} g(n) \leq \alpha g(n) \leq \underbrace{2\alpha}_{c_2} g(n)$$

$n_0=1$

$n \geq 1$

2018 2-)

a-) $2\log_2(n+65)^6$ Θ cinsinden?

$12\log_2(n+65) \quad n_0 \geq 64$

$c_1 \cdot lg(n) \leq 12\log_2(n+65) \leq \underbrace{24}_{c_2} lg(n)$

$\in \Theta(\log_2 n)$

b-) $\frac{n(n-1)(2n+5)}{6}$ n^3 Θ cinsinden?

$$\lim_{n\to\infty} \frac{n(n-1)(2n+5)}{6n^3} = \lim_{n\to\infty} \frac{(n^2-n)(2n+5)}{6n^3}$$

$$= \lim_{n\to\infty} \frac{2n^3+3n^2-5n}{6n^3} = \lim_{n\to\infty} \left(\frac{1}{3}+0\right)$$

c-) $\sqrt{8n^8+8n^2+\sqrt{n}}$ Θ cinsinden? (n^4 tahmin)

$$\lim_{n\to\infty} \sqrt{\frac{8n^8+8n^2+\sqrt{n}}{n^8}} = \lim_{n\to\infty} \sqrt{8} = 2\sqrt{2}$$

n^4'e böldük kökten çıkalım diye n^8 yaptık

d-) $\underbrace{2n\,lg(2n+2)^3}_{6n\,lg(2n+2)} + (n^2+2)^2 lg n$

$\underbrace{6n\,lg\,2n}_{12n\,lg\,n} \cdot \underbrace{lg\,2}_{1}$

$\underbrace{c_1}\, n\log_2 n \leq \quad \leq \underbrace{24}_{c_2} n\,lg\,n$

$(n^4+4n^2+4)\,lg n$

$\Theta(x) \quad \Theta(y)$

$\Theta(x \cdot y)$

$n^4 \quad lg n$

$\Theta(n^4 lg n)$

Bu daha büyük olduğundan çözüm bu

e-) $8^n+9^{2^n}$

$9^{2^n} \leq 8^n+9^{2^n} \leq 2 \cdot 9^{2^n}$

$c_1=1 \quad c_2=2$

$n \geq 1 \quad \in \Theta(9^{2^n})$

2018 3-)

codeforces

euler

$T(n) = 2T(n/2) - 2$, $T(1) = 8$

$n = 2^k$

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

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

$T(2^{k-2}) = 2T(2^{k-3}) - 2$

$T(2^k) = 2[2(2T(2^{k-3})-2)-2]-2$

$= 2^i T(2^{k-i}) - 2^{i-1} \dots 2^1$

$= 2^k \underbrace{T(1)}_{8} - 2^{k-1} \dots 2^1 = 2^{k+3} - \underbrace{2^{k-1} \dots 2^1}$

$2^{k+3} - (2^{k+1} - 2)$

$2^{k-i} = 1$

$k - i = 0$

$k = i$

$\boxed{2^k = n \quad \lg n = k}$

$2^{\lg n + 3} - 2^{\lg n + 1} - 2$

$T(n) = 8n - (2n - 2) = 6n + 2 \in \Theta(n)$

?? o o

for i = 0 → n

for J = 0 to n

sor

if Cevap = evet

break

else

Ünlü = A

index ? = i

A B C D

hayır hayır hayır → break

evet

! A

for i = 0 → n

sor A(i) ary biliyormusun

if hayır

count ++

if count == 0

(ünlü A'dır)

else go to başa

Ünlü Kişi Algoritması (3(n-1))

```
Celebrity (L1 [1 ... n])
  L2 = ?
  while (L1 size() >= 2)
    A = L1 get (0)
    B = L1.get (1)
    if (A.knows (B))
      L1.remove (A)
      L2.push (A)
    else
      L1.remove (B)
      L2.push (B)
  while (L2size() >= 2)
    if (A.knows (B))
      L1.remove (A)
      L2.push (A)
      C = L1.get (0)
    else
      L1.remove (B)
      L2.push (A)
      c = L1.get (0)
  for (item in L2)
    if (item.knows (c))
      return -1
for (item in L2)
  if (c.knows (item))
    return -1
```

3

Algoritmalar

16.04.19

Hafta-10

Sahte Para Problemi: Elimizde eşit kollu terazi ve n tane para var. Bir tanesi sahte para ve hafif. Sahte parayı nasıl buluruz?

NOT! Decrease and Conquer uygulayarak 3'e böleriz.

Eğer bölmezsek; $T(n) = T(\lfloor n/2 \rfloor) + 1 \Rightarrow \in \Theta(\log_2 n)$

Eğer bölersek;

$$T(n) = T(\lfloor n/3 \rfloor) + 1 \qquad T(3) = 1$$
$$T(n/3) = T(n/9) + 1$$
$$T(n/9) = T(n/81) + 1 \qquad \in \Theta(\log_3 n)$$
$$T(n) = T(n/3) + 1 + 1 + 1$$
$$= T(n/3^i) + i \qquad n/3^i = 3$$
$$\boxed{\log_3 n} \qquad 3^{i+1} = n$$
$$\log_3 n = i + 1$$

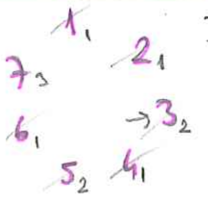
★ $\log_3 n$, $\log_2 n$'den daha hızlıdır. Peki kaç kat hızlıdır?

$$\frac{\log_2 n}{\log_3 n} = \frac{\log_3 2 \log_2 n}{\log_3 n} = \frac{\log_2 3 \cancel{\log_3 n}}{\cancel{\log_3 n}} = \log_2 3 \cong 1{,}6 \text{ kat}$$

Rus Köylü Problemi:

n çift ise	n tek ise
$n.m = \frac{n}{2} . 2m$	$n.m = \frac{(n-1)}{2} . 2m + m$

Josephus Problemi:



$J(7) = 7$

1. asker, 2.'yi öldürdü

3. asker, 4.'yü öldürdü

$J(11) = 7$

$J(11) = 2J(5) + 1$

$\underbrace{J(5)}_{3} = 2\underbrace{J(2)}_{1} + 1$

$J(3) = 3$

$2J(3) + 1 = J(7)$

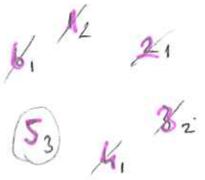
$2J(k) + 1 = J(2k+1)$

$$\boxed{\begin{array}{l} J(2k) = 2J(k) - 1 \\ J(2k+1) = 2J(k) + 1 \end{array}}$$

$J(12) = 9$

$J(12) = 2J(6) - 1$

$\underbrace{J(6)}_{5} = 2\underbrace{J(3)}_{3} - 1$



$\begin{pmatrix} 1 \to 1 \\ 3 \to 2 \\ 5 \to 3 \end{pmatrix}$ $J(6) = 5$

$f(i) = 2f(i) - 1$

küçük çemberde index

$J(6) = 2J(3) - 1$

$J(2k) = 2J(k) - 1$

1

Selection Problemi:

* k'ıncı en küçük elemanı bulmak için.
  - Bir pivot seçtik
  - Pivottan büyük olanları bir tarafa, küçük olanları bir tarafa attık.

$$\overbrace{a_{i_1} \;\ldots\; a_{i_{(s-1)}}}^{s>k} \quad p_k \quad \overbrace{a_{i_{(s+1)}} \;\ldots\; a_i}^{s<k}$$

k'ıncı — s=k ise — (k-s)'inci

Lomuto Partition

```
p ← A[l]
s ← l
for i ← l+1 to r
    if A[i] < p
        s ← s+1;
        swap A[s],A[i]
swap (A[l], A[s])
return s
```

0 1 2 3 4 5 6 7 8
[$\boxed{9}$ 10, 1, 5, 8, -1, 7, 16, 3]
s i

p = 9 A[s] = 10
A[i] = 10 A[i] = 1 > swap

9 1 10 5 8 -1 7 16 3

9 1 5 10 8 -1 7 16 3

9 1 5 8 10 -1 7 16 3

9 1 5 8 -1 10 7 16 3

9 1 5 8 -1 7 10 16 3

9 1 5 8 -1 7 3 16 10

s=6
pivotun olması gereken yer

3 1 5 8 -1 7 9 16 10

7. en küçük eleman

NOT ! s+1 değeri pivottan büyük olan ilk elemanı tutuyor.

Quickselect

```
s ← LomutoPartition(A[l...r])
if s = k-1
return A[s]
else if s > l+k-1
Quickselect (A[l... s-1],k)
else
Quickselect (A[s+1....r], k-1-s)
```

---

0 1 2 3 4 5 6 7 8
A = $\boxed{4}$ 1 10 8 7 12 9 2 15 k = 5

4 1 2 8 7 12 9 10 15 } LomutoPartition

2 1 4 (8 7 12 9 10 15)

s=2 l=3 r=8

Quickselect içinde else koşulunu uyguladık

$\boxed{8}$ 7 12 9 10 15

7 8 12 9 10 15

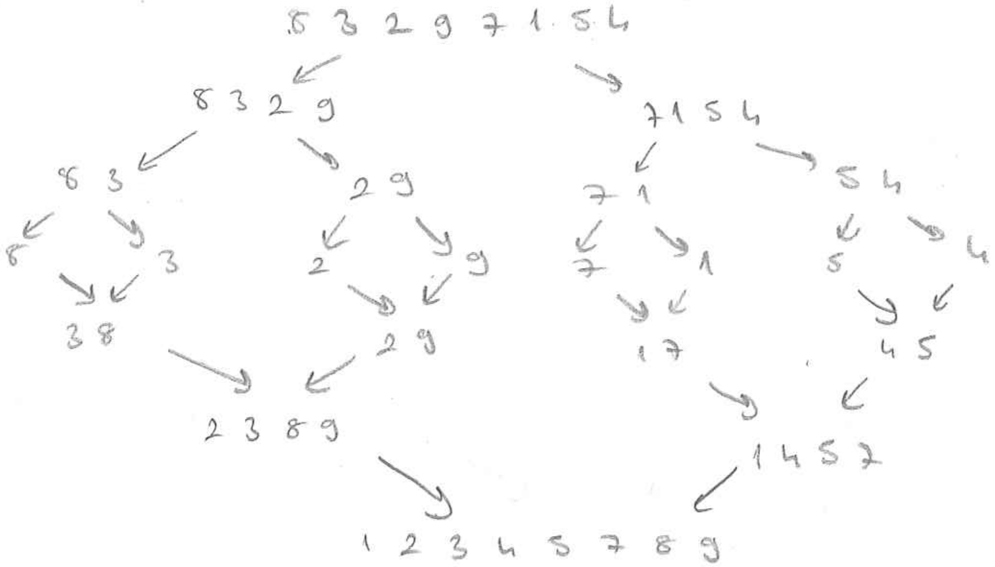
s=4

s = k-1
s = 4 ✓ 8

Divide and Conquer

Yaklaşım: (1) Problemi parçalara böl (2) Tüm parçaları özyineli şekilde çöz

Merge Sort



Algoritmalar

30.04.19
Hafta 12

$T(n) = aT(n/b) + f(n)$, öyle ki $f(n) \in \Theta(n^d)$, $d \geq 0$

Master Teoremi: If $a < b^d$, $T(n) \in \Theta(n^d)$
If $a = b^d$, $T(n) \in \Theta(n^d \log n)$
If $a > b^d$, $T(n) \in \Theta(n^{\log_b a})$

Örn: $T(n) = \underset{\uparrow a}{4}T(n/\underset{\uparrow b}{2}) + \underset{\downarrow}{n} \Rightarrow T(n) \in ?$

$$\Theta(n^{\log_2 4}) = \Theta(n^2)$$

Örn: $T(n) = 4T(n/2) + n^2 \Rightarrow T(n) \in ?$

$$\Theta(n^2 \log n)$$

Örn: $T(n) = 4T(n/2) + n^3 \Rightarrow T(n) \in ?$

$$\Theta(n^3)$$

Mergesort

* Array'i iki eşit parçaya böl
* Parçaları B ve C'ye kopyala
* B ve C'yi özyinelı şekilde sırala
* B ve C'yi A'ya merge et

Merge Algorithm

```
i = 0 ; j = 0 ; k = 0 ;
while i < p and j < q
    if B[i] ≤ C[j]
        A[k] = B[i]
        i = i+1
    else A[k] = C[j]
        j = j+1
    k = k+1
if i = p
    copy C[j...q-1] to A[k...p+q-1]
else copy B[i...q-1] to A[k...p+q-1]
```

⟹ En kötü durumda n kez döner (Bir A'dan bir B'den aldığımız durum)

Mergesort Algorithm

```
if n > 1
    copy A[0...[n/2]-1] to B[...[n/2]-1]
    copy        "        to C[...[n/2]-1]
    Mergesort(B[0...[n/2]-1])
    Mergesort(C[0...[n/2]-1])
    Merge(B,C,A)
```

NOT! Merge 2 sıralı listeyi 1 sıralı liste haline getirmek için kullanılır.

Mergesort Örneği

7/2 = 3

3 - 1 = 2

0, 1, 2. indexleri ilk kısma at

n = 7

Mergesort'ta en kötü ve en iyi durum aynıdır.

Her problemi sözdüğümüz sıra 2 kat sayısı

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

Merge algoritmasının zaman karmaşıklığı

$\in \Theta(n \log n)$

Quicksort

if $l < r$

  s = Partition (A[l...r])

  Quicksort (A[l....s-1])

  Quicksort (A[s+1...r])

Partition

p	all are $\le p$	$\ge p$	...	$\le p$	all are $\ge p$

$\rightarrow i$ $j \leftarrow$

p	all are $\le p$	$= p$	all are $\ge p$

$\rightarrow i = j \leftarrow$

p	all are $\le p$	$\le p$	$\ge p$	all are $\ge p$

$j \leftarrow$ $\rightarrow i$

* $i > j$ olduğu için işlem bitti pivotu yerine yerleştirdik

Hoare Partition

$p = A[l]$

$i = l$

$j = r+1$

repeat

  repeat $i = i+1$ until $A[i] \ge p$

  repeat $j = j-1$ until $A[j] \le p$

until $i \ge j$ // pivottan büyük bir eleman geldi

swap (A[i], A[j])

Quick Sort Best Case

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

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

Quick Sort Worst Case

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

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

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

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

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

Quick Sort Avegare Case

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

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

$$n T(n) = 2(T(0) + T(1) + \ldots + T(n-1)) + n(n+1)$$

$$(n-1) T(n-1) = 2(T(0) + \ldots + T(n-2)) + n(n-1)$$

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

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

$$\frac{T(n)}{n+1} = \frac{T(n-3)}{n-2} + \frac{2}{n+1} + \frac{2}{n} + \frac{2}{n-1} \ldots \frac{T(n)}{n+1} = 2 \sum_{k=3}^{n+1} \frac{1}{k} \quad \boxed{\frac{T(n)}{n+1} \le 2 \log(n+1)}$$

Büyük Tamsayıları Çarpma İşlemi:

Örn: 23 ve 14'ü çarp

$23 = 2.10^1 + 3.10^0 \qquad 14 = 1.10^1 + 4.10^0$

$(2.10^1 + 3.10^0) * (1.10^1 + 4.10^0)$

$= (2*1)10^2 + \underbrace{(3*1 + 2*4)}10^1 + (3*4)10^0$

* Normal çarpımda 2^2 4 çarpımda hesaplanıyor.
* 3 farklı çarpma işlemiyle hesapladık.

$(3*1 + 2*4) = (2+3)*(1+4) - (2*1) - (3*4)$

bunlar hesaplanmıştı

Yaklaşım: $a = a_1 a_0, \quad a = a_1 10^{n/2} + a_0$

$b = b_1 b_0, \quad b = b_1 10^{n/2} + b_0$

$c = a * b = c_2 10^2 + c_1 10^1 + c_0$

$c_2 = a_1 * b_1 \rightarrow$ İlk hanelerin çarpımı

$c_0 = a_0 * b_0 \rightarrow$ son " "

$c_1 = (a_1 + a_0) * (b_1 + b_0) - (c_2 + c_0)$ ortanca hanelerin sonucu

Örn: $\overset{a_1}{21}\overset{a_0}{01} \times \overset{b_1}{11}\overset{b_0}{30}$

$C_2 = a_1 b_1 * (10^4)$

$C_0 = 01 * 30$

$c_1 = (21 + 01) * (11 + 30) = (c_2 + c_0)$

! 9 tane çarpma işlemi yaptık !

$(21*11)$	$(01*30)$	$(22*41)$
$c_2' = 2*1 = 2$	$c_2' = 0$	$c_2' = 8$
$c_0' = 1$	$c_0' = 0$	$c_0' = 2$
$c_1' = (3*2 - 2 - 1)$	$c_1' = 1*3 - 0 = 3$	$c_1' = 4*5 - 8 - 2 = 10$
$c_1' = 3$	$c_2' c_1' c_0' \Rightarrow 30$	$\underline{902}$
$\Downarrow$ 231		

$902 - 231 - 30 = \underline{641}$

$231 * 10^4 + 641 * 10^2 + 30 = \underline{2374130}$

$M(n) - 3M(n/2)$

$M(2^k) = 3M(2^{k-1}) = 3[3M(2^{k-2})] = 3^2 M(2^{k-2})$

$= \dots = 3^i M(2^{k-i}) = \dots = 3^k M(2^{k-k}) = 3^k$

Matris Çarpımı:

NOT ! Strassen'in Matris Çarpımı

Ödev Verilecek

ALGORİTMALAR

07.05.19

Hafta-13

Element Uniqueness

* Bir array içerisindeki bir elemanın biricik olduğunu kontrol etmek

Bu algoritmayı brute force ile $\Theta(n^2)$'de çözeriz. İyileştirmek için ne yapılabilir?

* $\Theta(n\log n)$ zaman karmaşıklığıyla merge sortta sıraladık. Daha sonra bir sağındakiyle kontrol ettik. Çözüm en kötü durumda "$n\log n + n$"dir. Yani zaman karmaşıklığı $\Theta(n\log n)$

1 2 3 4 5 5 ~~6~~

Computing Mode

* Bir listede en çok görülen elemanı bulma

Brute force ile $\Theta(n^2)$'de çözeriz. Bu algoritmayı daha iyi çözmek için önce sıralayıp sonra aşağıdaki algoritmayı uygularız. Zaman karmaşıklığı "$n\log n + n$"dir. $\Theta(n\log n)$

1 1 1 2 2 2 2 4 5 6

```
while i ≤ n-1
    runlenght = 1
    runvalue = A[i]
    while i+runlenght ≤ n-1 and A[i+runlenght] = runvalue
        runlenght = runlenght + 1
    if runlenght ≥ mode fre
        mode fre = runlenght
        modevalue = runvalue
    i = i + runlenght
return modevalue
```

en kötü durum → Bu algoritma için $\Theta(n)$

Searching Problem

* n elemanlı bir arraydeki v değerini bulma algoritması

Önce sıralayıp sonra "Binary Search Algo." uygularsak algoritmayı iyileştirmiş oluruz.

Gaussian Elimination

* $$\begin{aligned} 2x_1 - x_2 + x_3 &= 1 \\ 4x_1 + x_2 - x_3 &= 5 \\ x_1 + x_2 + x_3 &= 0 \end{aligned} \Rightarrow \begin{bmatrix} 2 & -1 & 1 & 1 \\ 4 & 1 & -1 & 5 \\ 1 & 1 & 1 & 0 \end{bmatrix} \to \begin{bmatrix} 2 & -1 & 1 & 1 \\ 0 & 3 & -3 & 3 \\ 0 & 3/2 & 1/2 & -1/2 \end{bmatrix} \to \begin{bmatrix} 2 & -1 & 1 & 1 \\ 0 & 3 & -3 & 3 \\ 0 & 0 & 2 & -2 \end{bmatrix}$$

Gaussian Elimination

```
for i=1 to A[i,n+1] = b[i]
for i=1 to n-1
    for j=i+1 to n
        for k=i to n
            A[j,k] = A[j,k] - A[i,k] * A[j,i] / A[i,i]
```

$\in \Theta(n^3)$

* A[i,i] = 0 ise → Hata alırız.

* Eğer A[i,i] çok küçük ise → Hata alırız.

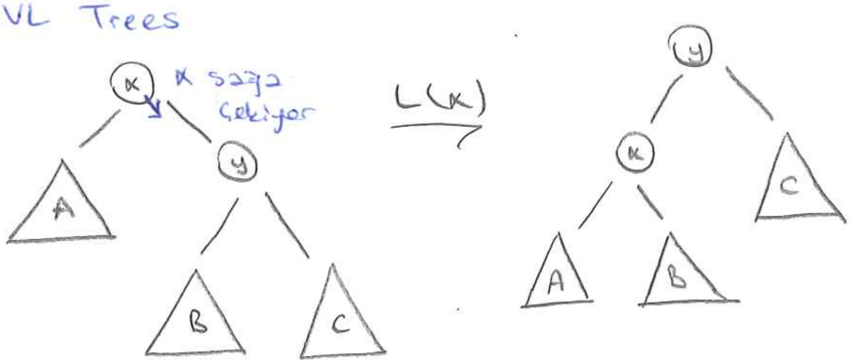
Better Gaussian Elimination

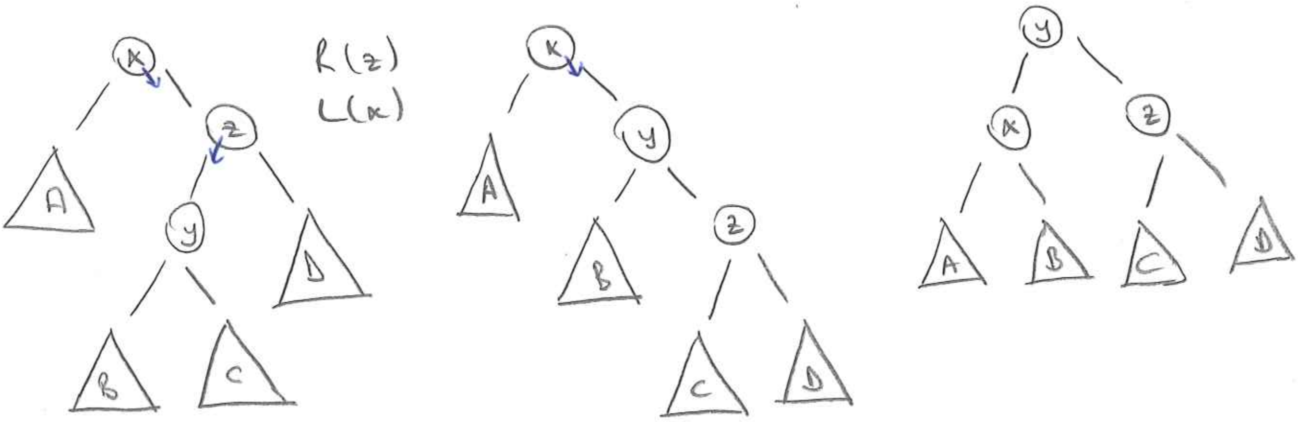
```
for i=1 to A[i,n+1] = b[i]
for i=1 to n-1
    pivotrow = i
    for j=i+1 to n
        if |A[j,i]| > |A[pivotrow,i]| pivotrow = j
    for k=i to n+1
        swap (A[i,k], A[pivotrow,k])
    for j=i+1 to n
        temp - - - - - - - -
```

* Balanced Search Trees

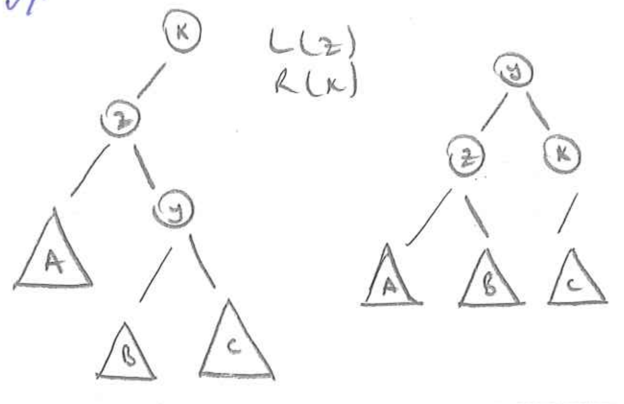
* Binary Search Tree'de ağacın derinliğinden kaynaklanan verimsizliği çözüyoruz.

AVL Trees

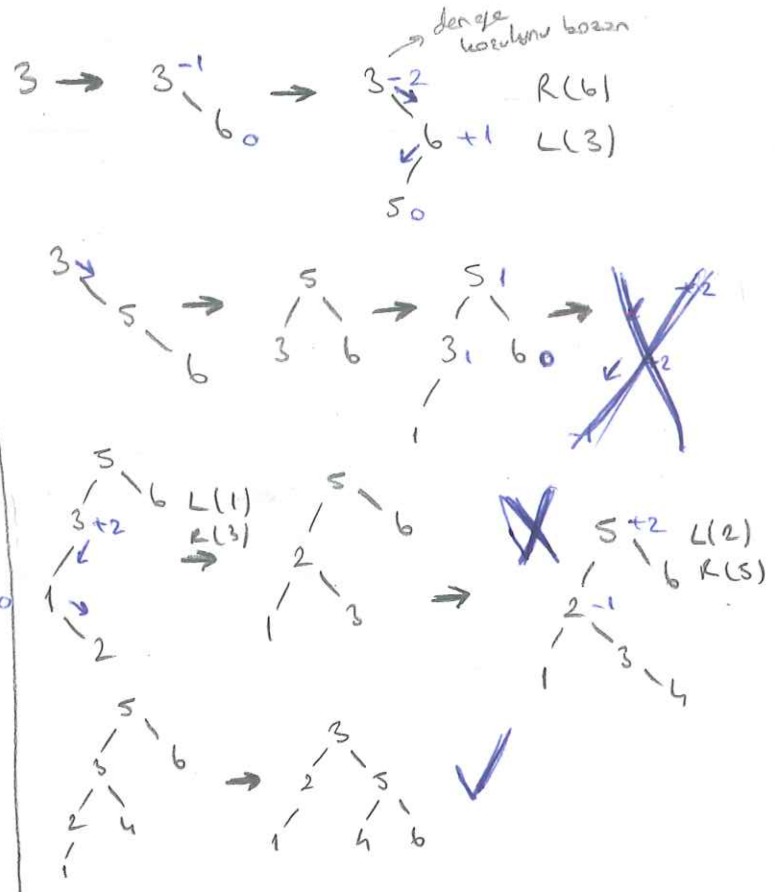




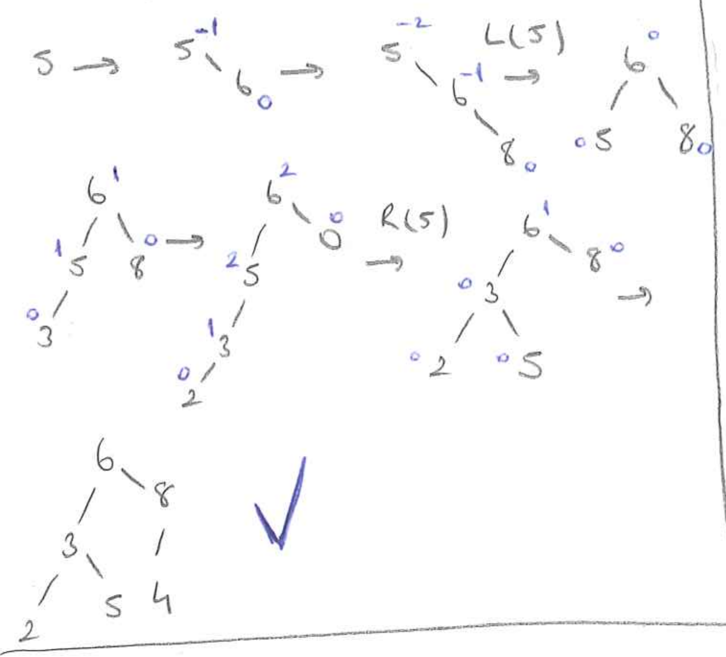
#



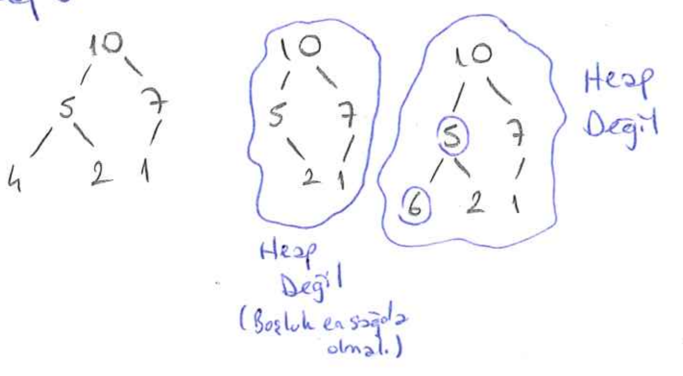
Örn: 3, 6, 5, 1, 2, 4 AVL ağacını oluştur.



Örn: 5, 6, 8, 3, 2, 4



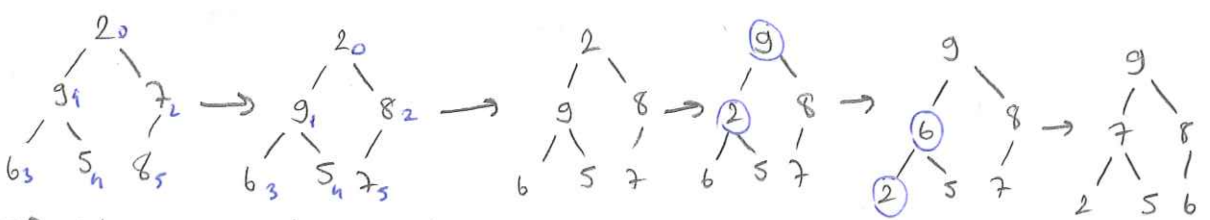
Heaps



NOT ! 2 koşul var:

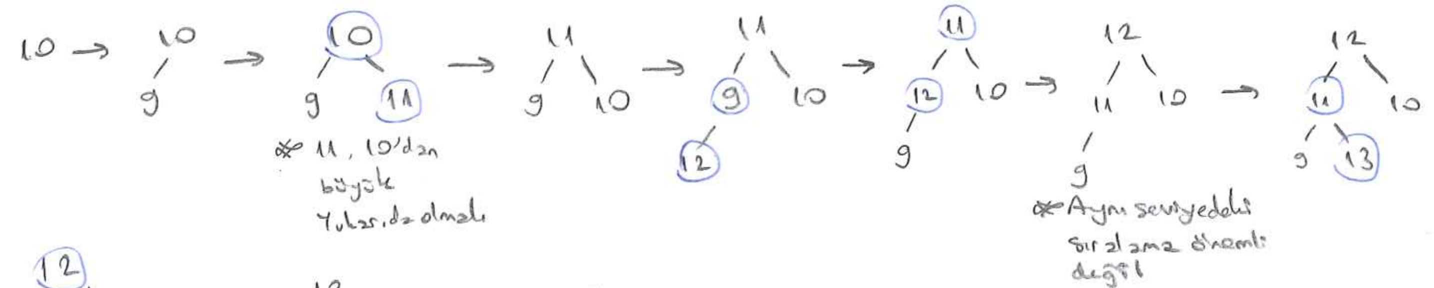
1. Boşluk olacaksa en sağda olmalı
2. Leafler roottan küçük olmalı

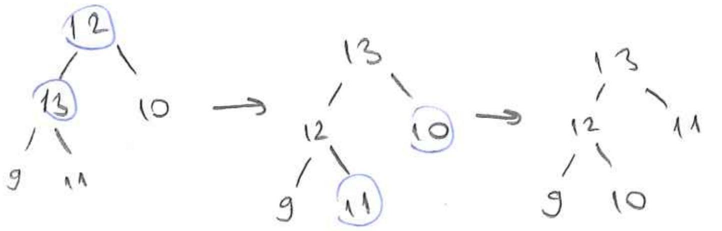
Örn: 2, 9, 7, 6, 5, 8 Heap ağacını oluştur.



* Direkt ağacı oluştur
* Parent özelliğine sahip en yüksek indise bak
* 9'da sıkıntı yok

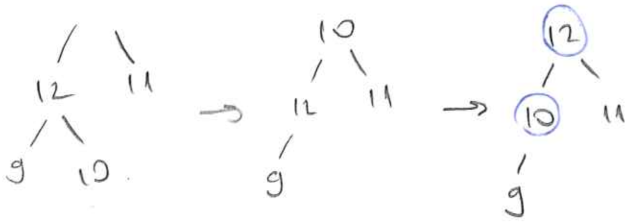
Örn! 10, 9, 11, 12, 13 Heap ağacını çiz.



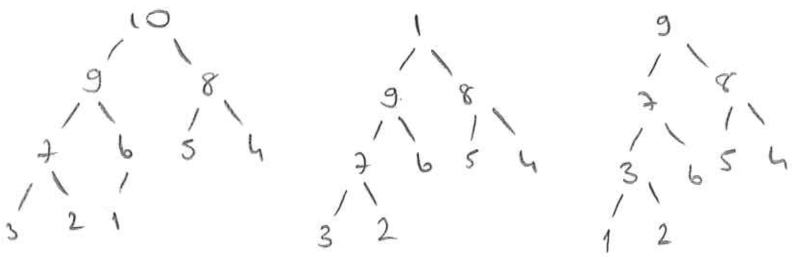


|13, 12, 11, 9, 10|

* Eğer root'u silmek istersek. Leafların en sağındaki elemanı roota koyarız.



Örn: |10, 9, 8, 7, 6, 5, 4, 3, 2, 1| heapinde 10'u silelim.



Heap Sort

* Sürekli silerek sıralama oluşturabiliriz. $\Theta(n\log n)$

Horner's Rule

* Polinomu hesapla $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$

Sürekli x parantezine alıp dönüştürüyoruz.

$P(x) = 2x^4 - x^3 + 3x^2 + x - 5$

$P(x) = x(2x^3 - x^2 + 3x + 1) - 5$

$= x(x(2x^2 - x + 3) + 1) - 5$

$= x(x(x(2x-1)+3)+1) - 5$

	2	-1	3	1	-5
$x=3$	2	$3.2+(-1) = 5$	$3.5+(3) = 18$	$3.18+1 = 55$	$3.55+(-5) = 160$

– Strassen'in Algoritması –

$$\begin{bmatrix}1&0&2&1\\4&1&1&0\\0&1&3&0\\5&0&2&1\end{bmatrix} * \begin{bmatrix}0&1&0&1\\2&1&0&4\\2&0&1&1\\1&3&5&0\end{bmatrix} \Rightarrow \left[\begin{array}{cc|cc}1&0&2&1\\4&1&1&0\\\hline 0&1&3&0\\5&0&2&1\end{array}\right] * \left[\begin{array}{cc|cc}0&1&0&1\\2&1&0&4\\\hline 2&0&1&1\\1&3&5&0\end{array}\right]$$

A   B   A   B

$$A_{11} = \begin{bmatrix}1&0\\4&1\end{bmatrix} \quad B_{11} = \begin{bmatrix}0&1\\2&1\end{bmatrix}$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$\begin{bmatrix}a&b\\c&d\end{bmatrix} \qquad \begin{bmatrix}e&f\\g&h\end{bmatrix}$$

$P1 = a*(f-h)$
$P2 = h*(a+b)$
$P3 = e*(c+d)$
$P4 = d*(g-e)$
$P5 = (a+d)*(e+h)$
$P6 = (b-d)*(g+h)$
$P7 = (a-c)*(e+f)$

$$\Rightarrow \begin{bmatrix}P6+P5+P4-P2 & P1+P2\\ P3+P4 & P1+P5-P7-P3\end{bmatrix}$$

$$A_{11} * B_{11} = \begin{bmatrix}0&1\\2&5\end{bmatrix}$$

$$\underbrace{A_{12} = \begin{bmatrix}2&1\\1&0\end{bmatrix} \quad B_{12} = \begin{bmatrix}0&1\\0&4\end{bmatrix}}_{A_{12} * B_{12} = \begin{bmatrix}0&6\\0&1\end{bmatrix}}$$

$$\underbrace{A_{21} = \begin{bmatrix}0&1\\5&0\end{bmatrix} \quad B_{21} = \begin{bmatrix}2&0\\1&3\end{bmatrix}}_{A_{21} * B_{21} = \begin{bmatrix}1&3\\10&0\end{bmatrix}}$$

$$\underbrace{A_{22} = \begin{bmatrix}3&0\\2&1\end{bmatrix} \quad B_{22} = \begin{bmatrix}1&1\\5&0\end{bmatrix}}_{A_{22} * B_{22} = \begin{bmatrix}3&3\\7&2\end{bmatrix}}$$

$C_{11} = A_{11} * B_{11} + A_{12} * B_{21}$
$C_{12} = A_{11} * B_{12} + A_{12} * B_{22}$
$C_{21} = A_{21} * B_{11} + A_{22} * B_{21}$
$C_{22} = A_{21} * B_{12} + A_{22} * B_{22}$

$$C_{11} = \begin{bmatrix}0&1\\2&5\end{bmatrix} + \begin{bmatrix}5&3\\2&0\end{bmatrix} = \begin{bmatrix}5&4\\4&5\end{bmatrix}$$

$$C_{12} = \begin{bmatrix}0&1\\0&8\end{bmatrix} + \begin{bmatrix}7&2\\1&1\end{bmatrix} = \begin{bmatrix}7&3\\1&9\end{bmatrix}$$

$$C_{21} = \begin{bmatrix}2&1\\0&5\end{bmatrix} + \begin{bmatrix}6&0\\5&3\end{bmatrix} = \begin{bmatrix}2&4\\0&11\end{bmatrix}$$

$$C_{22} = \begin{bmatrix}0&4\\0&5\end{bmatrix} + \begin{bmatrix}3&3\\7&2\end{bmatrix} = \begin{bmatrix}3&7\\7&7\end{bmatrix}$$

"
$$C = \begin{bmatrix}5&4&7&3\\4&5&1&9\\8&1&3&7\\5&8&7&7\end{bmatrix}$$
"

$$\boxed{T(n) = 7T(n/2) + n^2}$$

Beyza Kaynar
17253022

∝ Algoritmalar ∝

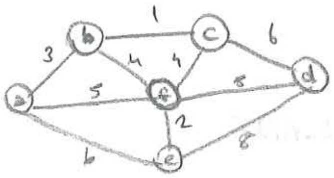
14.05.19

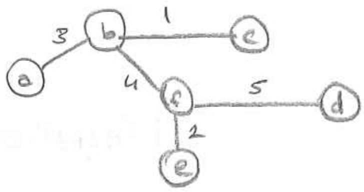
Hafta-14

Prim'in Algoritması:

∝ Greedy choise

for i = 1 to |V|-1 ⇒ Bu satırda greedy yaklaşım var
find a minimum-weight edge $e^* = (v^*, u^*)$ among all the edges (v, u) such that v is in V_T and u is in $V - V_T$
$V_T = V_T \cup \{u^*\}$
$E_T = E_T \cup \{e^*\}$
return E_T





1- a(-,∞) b(a,3)

2- b(a,3) c(b,1)
d(-,∞)
e(a,6)
f(b,4)

3- c(b,1) d(c,6)
e(a,6)
f(b,4)

4- f(b,4) d(f,5) e(f,2)

5- e(f,2) d(f,5)

6- d(f,5)

Zaman Karmaşıklığı
Matris ⇒ $(|V|)^2$
List ⇒ $|E| \log(V)$

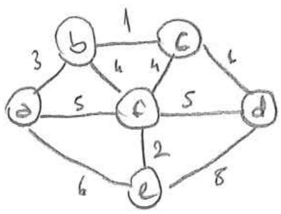
∝ Prim algoritması MST oluşturmaktadır.
∝ Tümevarım ile ispatlayalım.
* Sanki Prim Algo. yanlışmış gibi seçmemiz gereken edge'i seçmiyoruz.
* Yerine başka bir edge seçip optimum ağacı oluşturuyoruz.
* Sonra aslında seçmemiz gereken edge'i ekliyoruz. Bir döngü oluşuyor.
* Yerine tercih ettiğimiz edge'i çıkarıyoruz. Sonuç daha az maaliyetli olduğu için ispatlamış olduk.

Kruskal'in Algoritması:

∝ Greedy choise
∝ Edge'leri sıralıyoruz. Döngü oluşturmayacak şekilde minimumları seçiyoruz.

while ecounter < |V|-1 do
k = k+1
if $E_T \cup \{e_{ik}\}$ is acyclic
$E_T = E_T \cup \{e_{ik}\}$;
ecounter ← ecounter + 1
return E_T

Zaman Karmaşıklığı
$|E| . \log |E|$
$\log |V| . |E|$



bc	ef	ab	bf	cf	af	df	ae	cd	de
1	2	3	4	4	5	5	6	6	8

Encoding Text

	a	b	c	d	e	f	
freq	45	13	12	16	9	5	
fixed word	000	...	...			101	= 300
variable word	0	101	100	111	1101	1100	= 224

* Daha çok kullanılanları düşük bitle temsil ederiz.

(Önek Kodları)

* Ar(k)biri 0'la başlayamaz. Çünkü a zannedelir.

Huffmann Codes

∝ Karakterler ve frekansı veriliyor.

character	A	B	C	D	-
probability	0.35	0,1	0,2	0,2	0,15

$\boxed{\text{Slayttaki Huffman Örneğini İncele}}$

DİNAMİK PROGRAMLAMA

∝ Alt problemleri çözerken gözdüklerimizi tabloya kaydederiz. Tekrara düştüğümüzde tablodan çekeriz.

Örn: Elimizde paralar var. 5, 1, 2, 10, 6, 2 herhangi iki komşu parayı seçmeden maksimum parayı nasıl seçeriz.

$F(i) = \max\{F(i-2) + c_i, F(i-1)\}$

$F(0) = 0$

(5+2, 5) (10+5, 7) (2+15, 15)

C → 5 1 2 10 6 2

F → 5 5 7 15 15 17

****

C → 5 1 2 6 6 2

F → 5 5 7 11 13 13

Örn:

	1	2	3	4	5	6
1	0	0	0	0	1	1
2	0	1	1	2	2	2
3	0	1	1	3	3	4
4	0	1	2	3	3	5
5	1	1	2	3	4	5

Robot ya aşağı gidebiliyor ya da sağa kayabiliyor. Max kaç noktaya uğrar?

$C_{i,j} = \max\{C_{i-1,j}, C_{i,j-1}\} + C_{i,j}$

$C_{0,j} = 0$

$C_{0,i} = 0$

Örn:

K	K	K	K	K
K	K	K	K	K
K	K	K	K	K
K	K	K	K	K
K	K	K	K	K

Her bir bir lamba bir ampulü kapattığımızda tüm komşularını açarız. Tüm ampulleri aynı anda nasıl kapatırız?