ALGORITMA ANALİZİ
FİNAL ÇALIŞMA

Ex: $5x^3+2x^2+3 \leq c\,g(x)$

$|5x^3+2x^2+3| \leq |5x^3|+|2x^3|+|3x^3|$

$f(x) \leq 10 \cdot x^3 \Rightarrow c=10 \quad O(x^3) \quad g(x)=x^3$

$c.g'(x) \leq f(x) \Rightarrow c=5 \quad 5.x^3 \quad g'(x)=x^3$

Ex: $\frac{n.(n+1)}{2} \leq c.g(x) \Rightarrow n^2+n \leq n^2+n^2 \quad n \leq n^2$

Approximate Solving = Ya parametre yok, Ya da degiskenlerin ortalaması. Tahmin var.

Exact Solving = Matematiksel formül ile ifade edilir.

Algoritmalar nasıl tasarlanır ?
- Iterative mi Recursive mi karar verilir.
- Approximate solving mi exact solving mi karar verilir.

Algoritmanın Dogrulugunun İspatı
- Her durumda dogru sonuc vermeli
- Algoritmanın verdigi degerler ispat edilmeli (Matris formunda)

! Hata payı düşük olmalı - Parametre verilmeli - Ortalama - Kesin sonuç var mı? - Doğruluğu ispatlanıyor mu

Spanning Tree
× Tree has no cycles. × Spanning bütün köşelere uğrar, cycle oluşturmaz.

Prim Algoritması = Loop durumuna bakılır, The vertex bir kez seçilmeli, Yol hesabı yok.

Brute Force
- Problem statement & Definitions of the concepts (Durum & Tanım) - Ex: Bubble Sort - Selection Sort

Selection Sort
Minimum değer bul, başa getir:

```
for i ← 0 to n-2 do
    min ← i
    for j ← i+1 to n-1 do
        if A[j] < A[min] min ← j
    swap A[i] and A[min]
```

Bubble Sort

-Ilk indisten basla, bir sonraki indisten büyükse yer degistir: for i←0 to n-2 do
for j←0 to n-2-i do
if A[j+1] < A[j]
swap A[j] and A[j+1]

Ex :

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

<1,2,3,4> = 1+2+4+6 = 13

Insertion Sort

- Ikili kontrol yapar, bir sonraki indisten büyükse yer degistirir. Küçük en sola gelir.

```
for i←1 to n-1 do
    u ← A[i]
    j ← i-1
    while j ≥ 0 and A[j] > u do
        A[j+1] < A[j]
        j ← j-1
    A[j+1] ← u
```

2) Find the big-O notation of the algorithm given by $f(n) = 100n + 5$

a) For all $n \geq 5$

$f(x) \leq c.g(x)$

$100n + 5 \leq 100n + n$

$100n + 5 \leq 101n$

$g(x) = O(n)$

$c = 101$

b) For all $n \geq 1$

$f(x) \leq c.g(x)$

$100n + 5 \leq 100n + 5n$

$100n + 5 \leq 105n$

$g(x) = O(n)$

$c = 105$

Merge Sort

if $n > 1$

copy $A[0 \dots (n/2)-1]$ to $B[0 \dots (n/2)-1]$

copy $A[(n/2) \dots (n-1)]$ to $C[0 \dots (n/2)-1]$

Mergesort $(B[0 \dots (n/2)-1])$

Mergesort $(C[0 \dots (n/2)-1])$

Merge (B, C, A)

Transform & Conquer

Problem's Instance $\Rightarrow$ Simpler or Instance / Another Representation / or Another Problem's Instance $\Rightarrow$ Solution

Greedy Algorithms

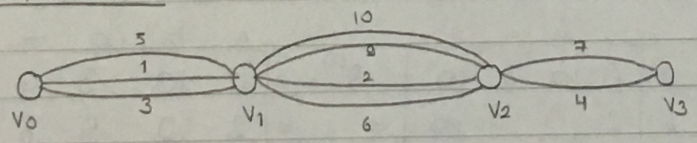
Change-Making

$d_1 = 25$ (quarter), $d_2 = 10$ (dime), $d_3 = 5$ (nickel), $d_4 = 1$ (penny)

$48 = 25 + 10 + 10 + 1 + 1 + 1 = (6)$

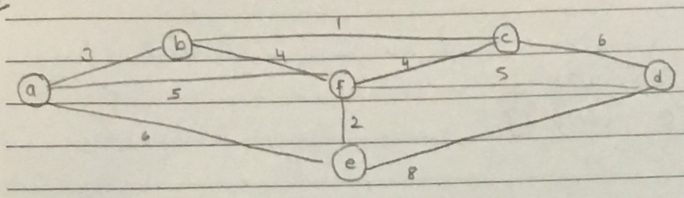
$d_1 = 20 \quad d_2 = 19 \quad d_3 = 5 \quad d_4 = 1 \qquad 24 = 20 + 1 + 1 + 1 + 1 = (5)$

Graph Problems

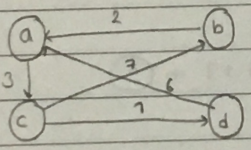


$1 + 2 + 4 = 7$

PRIM



Tree Vertices	Remaining Vertices	
a(-,-)	b(a,3), c(a,∞), d(-,∞), e(a,6), f(a,5)	b(a,3)
b(a,3)	c(b,1), d(-,∞), e(a,6), f(b,4)	d(b,5) c(b,7) e(d,9)
c(b,1)	d(c,6), e(a,6), f(b,4)	
f(b,4)	d(c,6), e(f,2)	
e(f,2)	d(f,5)	



$D^{(0)}$ =

	a	b	c	d
a	0	∞	3	∞
b	2	0	∞	∞
c	∞	7	0	1
d	6	∞	∞	0

⇒ $D^{(1)}$ =

	a	b	c	d
a	0	∞	3	∞
b	2	0	5	∞
c	∞	7	0	1
d	6	∞	9	0

$D^{(2)}$ =

	a	b	c	d
a	0	∞	3	∞
b	2	0	5	∞
c	9	7	0	1
d	6	∞	9	0

→ $D^{(3)}$ =

	a	b	c	d
a	0	10	3	4
b	2	0	5	6
c	9	7	0	1
d	6	16	9	0