AYRIK MATEMATİK 20.09.2017

Rosen, Discrete Mathematics and Its Applications

- Conjuction AND , $\wedge$
- Inclusive Disjunction, OR $\vee$
- Exclusive disjunction, XOR $\underline{\vee}$, $\oplus$
- Negation $\sim$, $\neg$
- Implication $\rightarrow$
- Double Implication $\leftrightarrow$

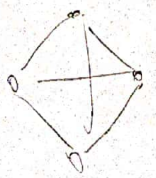
Logical Equivalence, $\longrightarrow (p \leftrightarrow q) \longrightarrow ((p \rightarrow q) \wedge (q \rightarrow p))$

Logical Equivalence, $\longrightarrow (p \rightarrow q) \longrightarrow \neg p \vee q$

Contrapositive, $\longrightarrow (p \rightarrow q) \longrightarrow (\neg p \longrightarrow \neg q)$

Tautology, $\longrightarrow$ Her zaman true $\longrightarrow (p \rightarrow (p \vee q))$

Contradiction, $\longrightarrow$ Her zaman false $\longrightarrow (p \wedge (\neg p))$



04.10.2019

4 Değişkenli

E/ $F(A,B,C,D) = \sum(0,5,7,9,10,13,15)$

$\bar{A}\bar{B}\bar{C}\bar{D}$

CD \ AB	00	01	11	10
00	0 1	4 0	12 0	8 0
01	1 0	5 1	13 1	9 1
11	3 0	7 1	15 1	11 0
10	2 0	6 0	14 0	10 1

$A\bar{C}D$

$2^3 2^2 2^1 2^0$
$A\ B\ C\ D$

Grup
$2^4 = 16$ X
$2^3 = 8$ X
$2^2 = 4$ ✓
$2^1 = 2$ ✓

Ortak olan B ve D
↓
BD

$A\bar{B}C\bar{D}$

$F_3(A,B,C,D) = A\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} + BD + A\bar{B}C\bar{D}$

E/

$\bar{B}\bar{C}\bar{D}$

CD \ AB	00	01	11	10
00	1	0	1	1
01	0	1	1	0
11	0	1	1	0
10	0	0	1	1

$A\bar{D}$

BD

$BD + A\bar{D} + \bar{B}\bar{C}\bar{D}$

E/

CD \ AB	00	01	11	10
00	1	1	1	1
01	1	0	0	1
11	1	0	0	1
10	1	1	1	1

$\bar{D}$

$\bar{B}$

$\bar{D} + \bar{B}$

E/

CD \ AB	00	01	11	10
00	1	1	1	1
01	1	1	0	0
10	0	0	0	0
11	1	1	1	1

$\bar{A}\bar{C}$

$\bar{D}$

$\bar{A}\bar{C} + \bar{D}$

E/ Bir öğrenci bir dönemde (max) 3 ders alabilir. Bir üst dönem yeni ders seçebilmesinin şartı en az 2'sinden geçmelidir. Bu durumun matematiksel modelini çıkartın.

indeks-1

	A	B	C	F
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

$F(A,B,C) = \sum(3,5,6,7)$

C \ AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$F_S(A,B,C) = AB + BC + AC = AB + C(A+B)$

String

$(ab)^*c$	a^*c^+
c	c
abc	cc
ababc	ccc
abababc	ac
	aaccc
	aaacc

→ Kleene Operator

11.10.2019

Bağıntı

$a = \{x, y\}$ $b = \{1, 2\}$

$a \times b = \{(x,1), (x,2), (y,1), (y,2)\}$

Bağıntı Özellikleri

★ A relation R on a set A is called Reflexive (yansıma) if $(x,x) \in R$ for every element $x \in A$

$A \times A \longrightarrow R : \{(1,1), (2,2), (3,3)\}$

	1	2	3
1	1	0	0
2	0	1	0
3	0	0	1

★ A relation R on a set A is called nonreflexive if $(x,x) \notin R$ for some element $x \in A$.

	1	2	3
1	0	0	1
2	1	0	0
3	1	1	1

★ A relation R on a set A is called irreflexive if $(x,x) \notin R$ for every element $x \in A$.

	1	2	3
1	0	1	1
2	1	0	1
3	0	0	0

★ A relation R on a set A is called symmetric (transpose) if $[(x,y) \in R$ whenever $(x,y) \in R]$ or $[(y,x) \notin R$ whenever $(x,y) \in R]$ or $(x=y)$; for $x, y \in A$

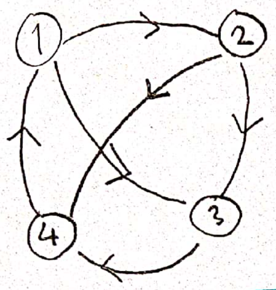
★ A relation R on a set A such that $(x,y) \in R$ and $(y,x) \in R$ only if $x=y$; for $x,y \in A$ is called antisymmetric. (x,y) var ken (y,x) olmamalı

★ A relation R on a set A is called transitive (geçişkenlik) if whenever $(x,y) \in R$ and $(y,z) \in R$ then $(x,z) \in R$ for $x,y,z \in A$

$\underbrace{(1,2)\,(2,3)}_{(1,3)}$

$X = \{1,2,3,4\}$ $(x,y) \in R$ $R: x \leq y$ $x,y \in X$

$R = \{(1,1),(2,2),(3,3),(4,4),(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$



	Ref	IrRef	NRef	Sym	Asym	Trans
$\{(2,2),(2,3),(2,4),(3,2),(3,3),(3,4)\}$	X	X	✓	X	X	✓
$\{(1,1),(1,2),(2,1),(2,2),(3,3),(4,4)\}$	✓	X	X	✓	X	✓
$\{(2,4),(4,2)\}$	X	✓	X	✓	X	X
$\{(1,2),(2,3),(3,4)\}$	X	✓	X	X	✓	X
$\{(1,1),(2,2),(3,3),(4,4)\}$	✓	X	X	✓	✓	✓
$\{(1,3),(1,4),(2,3),(2,4),(3,1),(3,4)\}$	X	✓	X	X	X	X

if $a = b^2$, $(a,b) \in R$ $A = \{1,2,3,4\}$

$R = \{(1,1),(4,2)\}$

Ref X
NRef ✓
IRef X
Sym X
Asym ✓
Trans ✓

Bağıntının Bileşkesi (Composition)

$R^1 = R$

$R^2 = R \circ R$

$R^3 = R^2 \circ R$

- - - - - - - - -

$R^n = R^{n-1} \circ R$

$R = \{(1,1), (2,1), (3,2), (4,3)\}$ için R^2 ve R^3 bul.

(2,1) (1,1)

(1,1) (2,1) (3,1)

$R^2 = R \circ R = \{(1,1)(2,1)(3,1)(1,2)\}$

$R^3 = R^2 \circ R = \{(1,1)(2,1)(3,1)(4,2)\}$

$R = \{(1,1),(2,1),(3,2)(4,3)\}$
$R = \{(1,1),(2,1),(3,2),(4,3)\}$ transitive bak

$R^2 = \{(1,1)(2,1)(3,1)(1,2)\}$

$R = \{(1,1)(2,1)(3,2)(4,3)\}$

Denklik Bağıntısı (Equivalence Relation)

X bir küme, R'de X üzerinde bir bağıntı olsun.

R bağıntısı üzerinde $\boxed{\text{reflexive, symmetric, transitive}}$ özellikleri mevcut ise bu bir denklik bağıntısı olup $X \Leftrightarrow R$ şeklinde gösterilir.

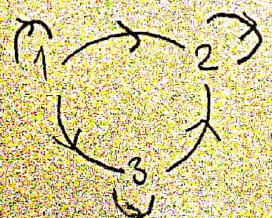
$X = \{\text{integers}\}$ ve X kümesi üzerinde tanımlı olan R bağıntısı da $xRy \Leftrightarrow x-y=5$ olarak verilsin. R'nin Equivalence Relation olup olmadığını göster.

$X = \{1, 6\}$, $R = \{(6,1)\} \longrightarrow$ IRef
Asym
Trans

$X = \{1,2,3,4,5,6\}$

$R = \{(1,1)(1,3)(1,5)(3,1)(3,3)(3,5)(5,1)(5,3)(5,5)(2,2)(2,6)(6,2)(6,6)(4,4)\}$

↓
Sym
Ref
Trans



$\underbrace{(1,1)(1,2)}_{(1,2)}$

Sıralama Bağıntısı (Partial Order Relation)

R: if x divides y (y/x) $(x,y) \in R$ $A = \{1,2,3,4\}$ $x,y \in A$

$R = \{(1,1)(2,2)(3,3)(4,4)(1,2)(1,3)(1,4)(2,4)\}$

Ref ✓
Asym ✓
Trans ✓

Partial order ✓

Hasse Diyagramları

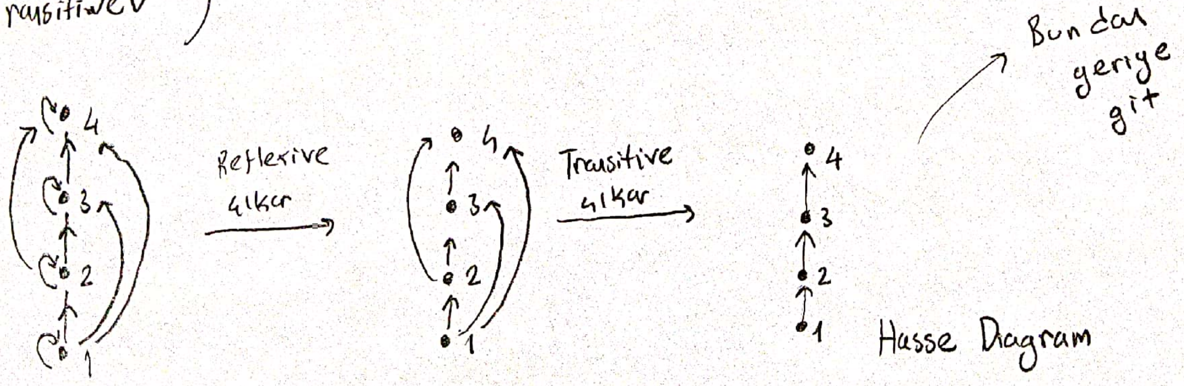
Sıralama bağıntısı özelliğini sağlarlar

$R: \{(a,b) \mid a \leq b,\ a \in X,\ b \in X\}$, $X = \{1,2,3,4\}$

$R = \{(1,1)(2,2)(3,3)(4,4)(1,2)(1,3)(1,4)(2,3)(2,4)(3,4)\}$

Reflexive ✓
Asm ✓
Transitive ✓

partial order



Hasse Diagram

Kapalılık (Closure)

reflexive yok
symmetric yok
transitive yok

bağıntının bu özelliklere sahip olabilmesini sağlama işlemidir.

* Transitive Closure
* Warshall Algorithm (by stephen warshall)

$W = M_R$
for $k=1, n$
  for $i=1, n$
    for $j=1, n$
      $w_{ij} = w_{ij} \vee (w_{ik} \wedge w_{kj})$
    end
  end
end

Elle çöz

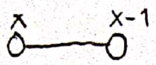
	1	2	3
1	1	1	0
2	0	1	0
3	1	0	0

$R = \{(1,1)\ (1,2)\ (2,2)\ (3,1)\}$

18.10.2019

Chromatic Polinoms

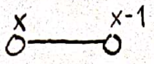
noktasal graf



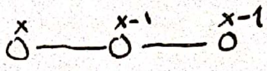
çizgisel graf

U Grafı

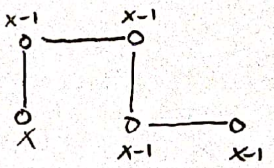
$U_n = x \cdot (x-1)^{n-1} \longrightarrow U_2 = x \cdot (x-1) \longrightarrow$



$\searrow U_3 = x \cdot (x-1)^2 \longrightarrow$



$\searrow U_5 = x \cdot (x-1)^4 \longrightarrow$

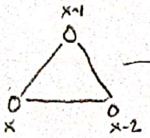


K. grafı

· her düğüm kendisi dışındaki düğümlere bağlıdır

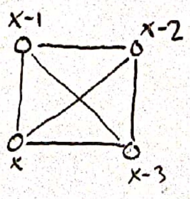
$K_3 \longrightarrow Z$ grafı (özel durum)

// polinomun 2. teriminin katsayısı grafın kenar sayısını verir.



$\longrightarrow K_3 = Z = x \cdot (x-1) \cdot (x-2)$

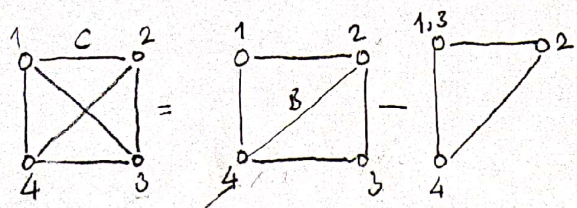
$K_n = \prod_{k=0}^{n-1} (x-k) \longrightarrow x \cdot (x-1) \cdot (x-2) \cdot (x-3) \longrightarrow$



$K_4 = (x^2 - x) \cdot (x^2 - 5x + 6) = x^4 - 6x^3 + 11x^2 - 6x$

- polinomun en yüksek dereceli terimin katsayısı 1'dir.
- polinomun derecesi graftaki düğüm sayısını verir.
- polinomun katsayıları toplamı sıfırdır.
- polinomda kesinlikle sabit sayı olmaz.

Grafın Sadeleştirilmesi

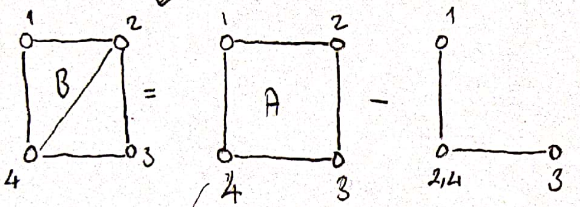


1,3 arasındaki path'i çıkar

1,3 üstüste gelecek şekilde katla

$$C = x(x-1)\cdot(x^2-4x+4) - x\cdot(x-1)\cdot(x-2)$$

$$C = x^4 - 6x^3 + 11x^2 - 6x$$

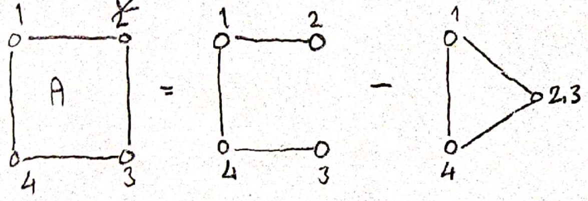


2,4 arası path'i çıkar

2,4 üst üste gelecek şekilde katla

$$B = x\cdot(x-1)\cdot(x^2-3x+3) - x\cdot(x-1)^2$$

$$B = x\cdot(x-1)\cdot(x^2-4x+4)$$



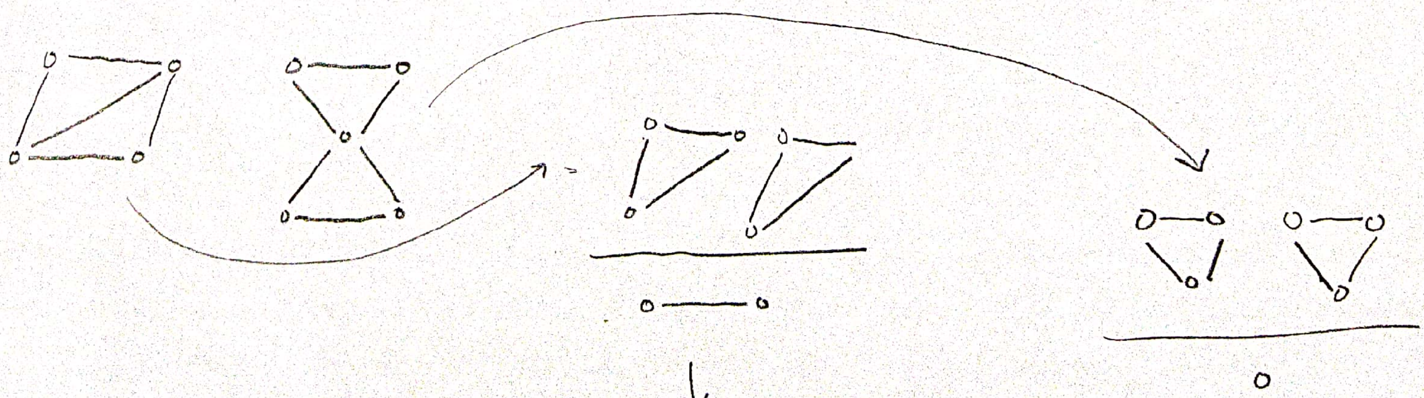
2,3 arası path'i kopar

2,3 üst üste katla

$$A = x\cdot(x-1)^3 - x\cdot(x-1)\cdot(x-2)$$

$$A = x\cdot(x-1)\cdot(x^2-3x+3)$$

Eğer 2 graf birbirinden noktasal veya çizgisel bir grafta ayrılıyorsa



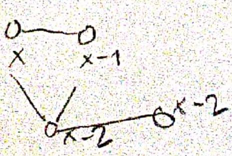
$$\downarrow$$

$$\frac{\cancel{x^2}\cdot(x-1)^2\cdot(x-2)^2}{x\cdot\cancel{(x-1)}}$$

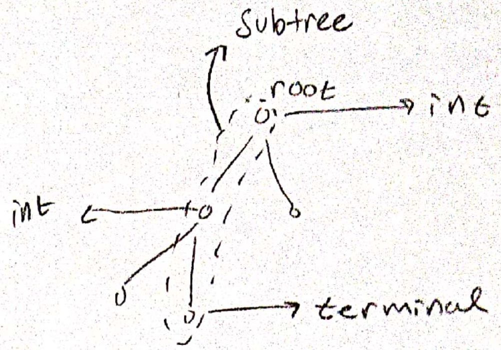
$$x\cdot(x-1)\cdot(x-2)$$

$$\downarrow$$

$$\frac{x^2\cdot(x-1)^2\cdot(x-2)^2}{x}$$



Trees



Dairesel yapı içermez

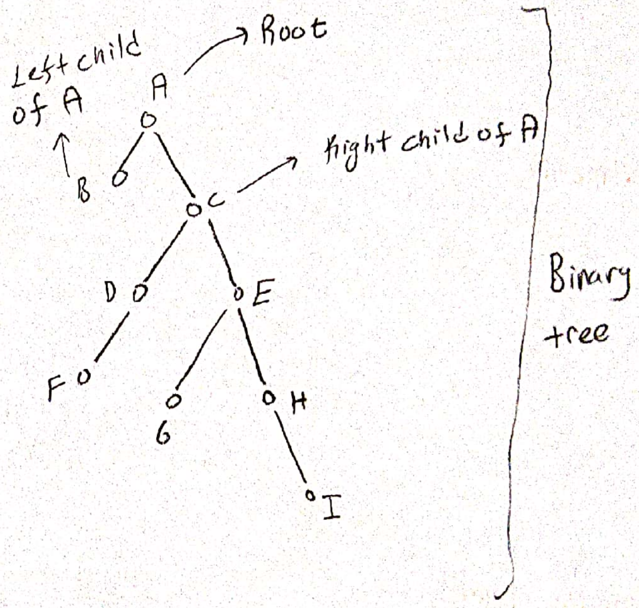
Bir node'dan diğerine tek yol var

$h = \max\{I(v)\}$

En az 1 çocuğu olan düğümler internal

Hiç çocuğu olmayanlar terminal

Ağaçların Karakteristiği



T → n düğümlü bir graf

a) T bir graftır

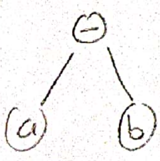
b) T bağlantılı ve acyclic

c) T bağlantılı ve n-1 kenara sahip

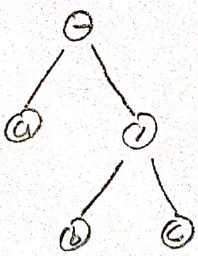
d T acyclic ve n-1 kenara =

Binary Tree- Matematiksel ifadelerin gösterimi

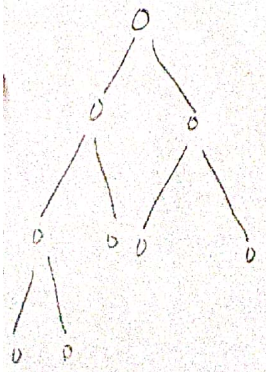
a- b



a- b/c



Tam ikili Ağaç (Full Binary Tree)



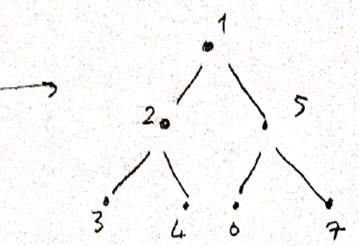
root node dışı her node 2 çocuğa sahip

t = uç düğüm sayısı , i = ara düğüm sayısı

- $\lg(t) \le h$
- $i+1$ = uç düğüm
- $2i+1$ = toplam düğüm

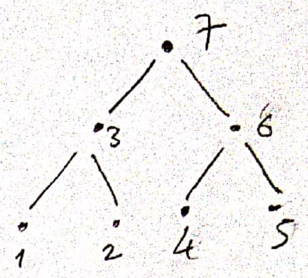
Tree Traversals

Pre-order →

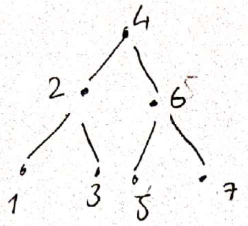


Post Order →

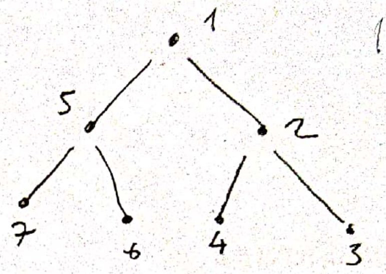
Post modernist sağ



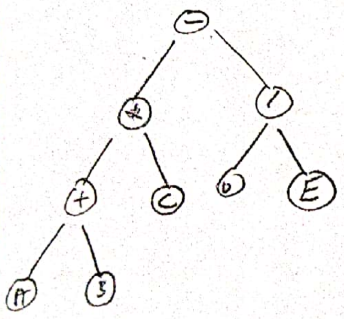
In-Order →



Reverse Post-Order →



Aritmetik



→ (A+B)*C - D/E infix

AB+C*DE/- postfix — sağdan sola

-*+ABC/DE prefix — soldan sağa

- (A+B)*C - D/E

(A+B)*C

Isomorphism:

Düğüm sayıları aynı

Kenar sayıları aynı

Karşılıklı düğüm derecesi aynı

Spanning Tree

G grafında bir T spanning tree

G grafının bir subtree'si

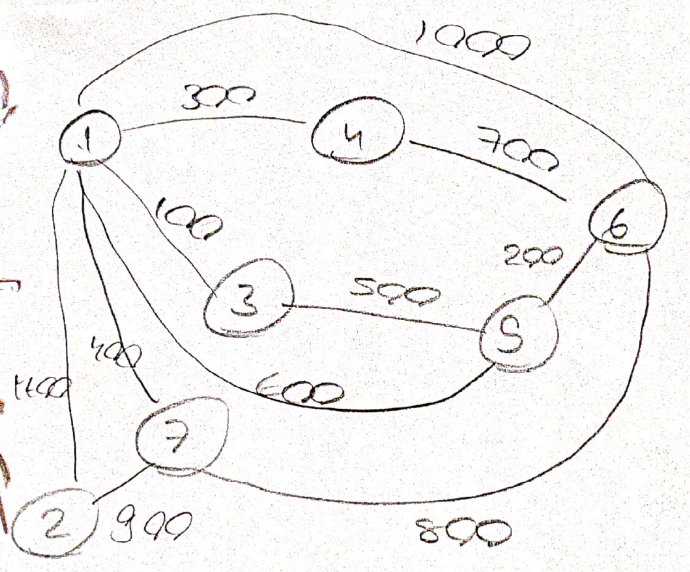
G grafının bütün düğümlerini içerir

Kruskal's Algorithm

Adım 1: en küçük ağırlıklı kenarı bul

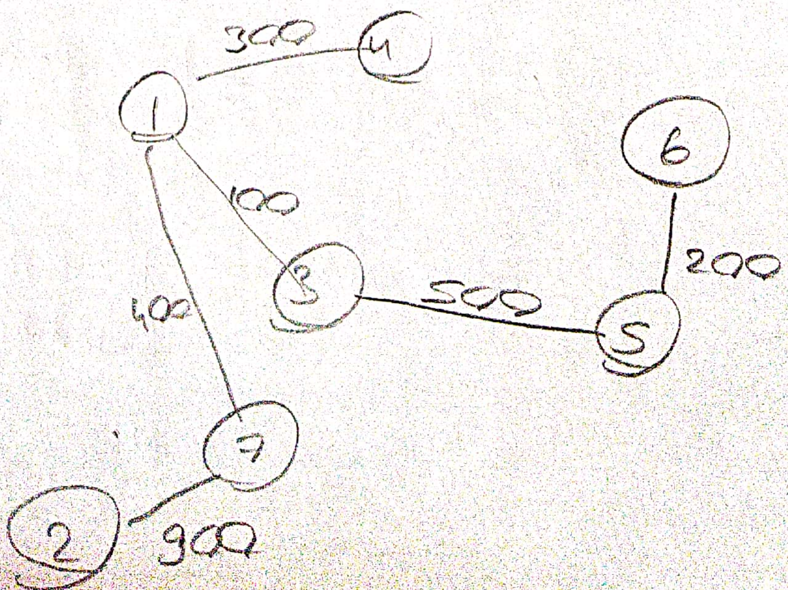
Adım 2: En küçük değere sahip bir sonraki kenarı bul ancak kapalı döngü oluşturmasın

Adım 3: Graftaki her bir düğümü dolaşana kadar adım 2 yap

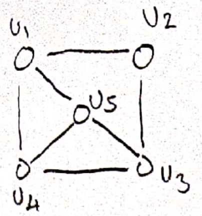


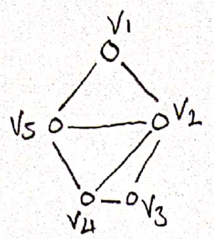
1 2 3

		1	2	3	4	5	6	7
x	1	M	1100	100	300	600	1000	400
X	2	1100	M	M	M	M	M	900
X	3	100	M	M	M	500	M	M
X	4	300	M	M	M	M	700	M
X	5	600	M	500	M	M	200	M
X	6	1000	M	M	700	200	M	800
X	7	400	900	M	M	M	800	M



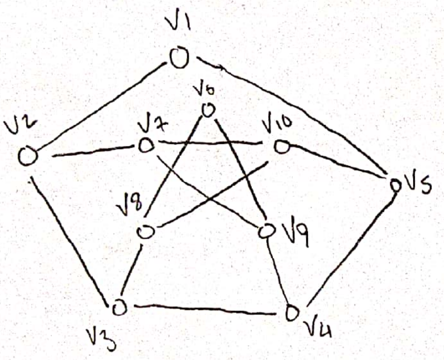
Graf isomorfizmi

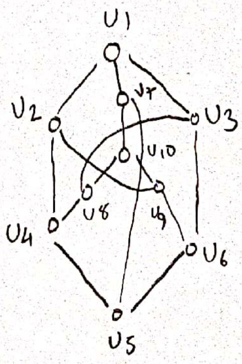




isomorfik değiller

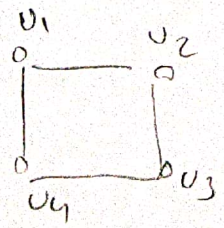
Düğüm = 5		Düğüm = 5
Kenar = 7	⟷	Kenar = 7
1 tane deg = 2 4 tane deg = 3	$\neq$ ⟷	2 tane deg = 2 2 tane deg = 3 1 tane deg = 4

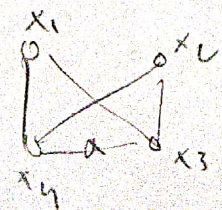




Kendisinin
ve komşularının
degree'si aynı mı?

Düğüm = 10		Düğüm = 10
Kenar = 15	⟷	Kenar = 15
10 tane deg = 3	⟷	10 tane deg = 3





derecenin çift olması lazım

E Graf → başladığın yere

E path → " " zorunlu değil

H cycle → her node'1 kez
kenar zorunlu değil

	x_1	x_2	x_3	x_4
x_1	0	0	1	1
x_2	0	0	1	1
x_3	1	1	0	0
x_4	1	1	0	0

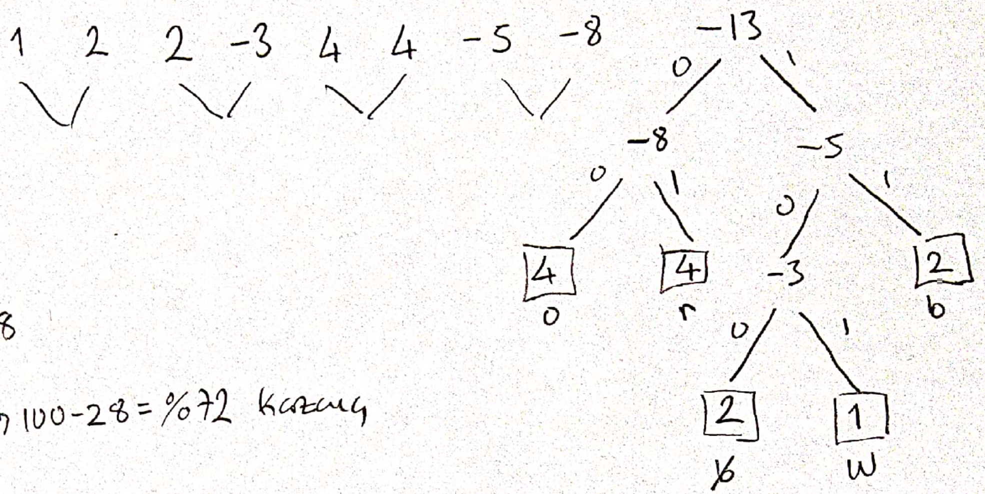
	U_1	U_2	U_3	U_4
U_1	0	1	0	1
U_2	1	0	1	0
U_3	0	1	0	1
U_4	1	0	1	0

Sınav Soruları

1) "borrow ɓ or ɓ rob"

b=2
o=4
r=4
w=1
ɓ=2

$13*8=104$ bit

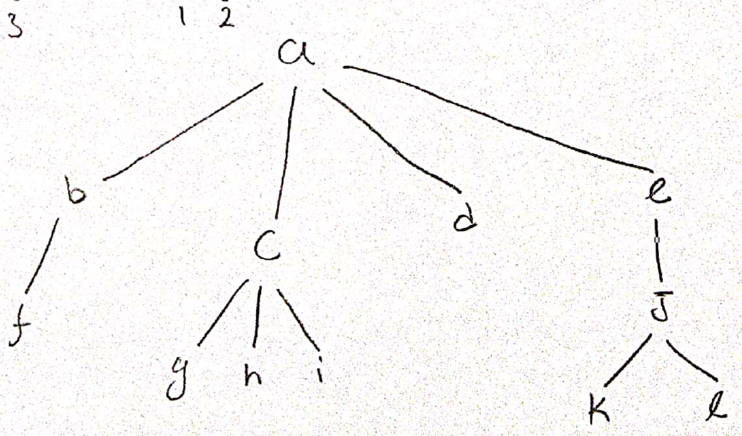


$\frac{29}{104} \rightarrow \%28$

↳ 100-28=%72 kazanç

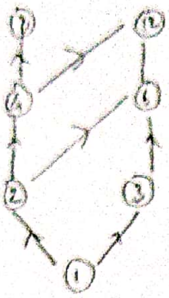
↳ o= 00 → 2*4=8
r= 01 → 2*4=8
b=11 → 2*2=4
ɓ=100 → 3*2=6
w=101 → 3*1=3
+
29 bit

2) a b f c g h i d e j k l
a → 4, b → 1, c → 3, e → 1, j → 2

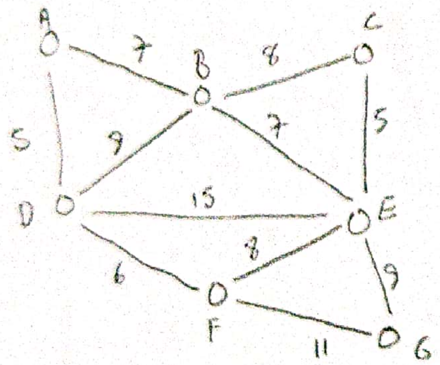


3) $A = \{1, 2, 3, 4, 6, 8, 12\}$ a b'yi böler. (a, b)

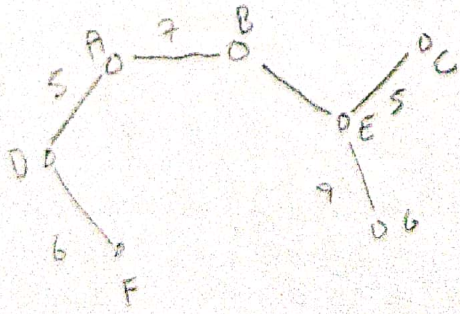
$R = \{(1,2)(1,3)(1,4)(1,6)(1,8)(1,12)(2,4)(2,6)(2,8)(2,12)(3,6)(3,12)(4,8)(4,12)(6,12)$
$\underline{(1,1)}\ \underline{(2,2)}\ \underline{(3,3)}\ \underline{(4,4)}\ \underline{(6,6)}\ \underline{(8,8)}\ \underline{(12,12)}\}$



4)



		A	B	C	D	E	F	G
2	A	(	$\boxed{7}$	m	5	m	m	m
4	B	7	m	8	9	$\boxed{7}$	m	m
6	C	m	8	m	m	5	m	m
1	D	$\boxed{5}$	9	m	m	15	$\boxed{6}$	m
5	E	m	7	$\boxed{5}$	15	m	8	$\boxed{9}$
3	F	m	m	m	6	8	m	11
7	G	m	m	m	m	9	11	m



AB ≠ 2A

5)

S_1: $a = x+y$

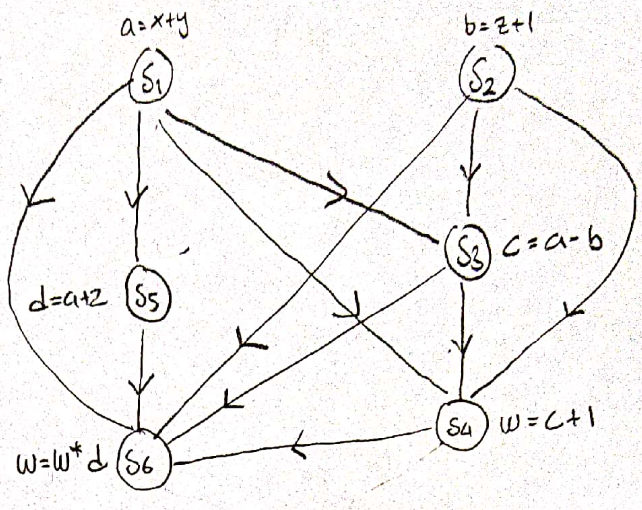
S_2: $b = z+1$

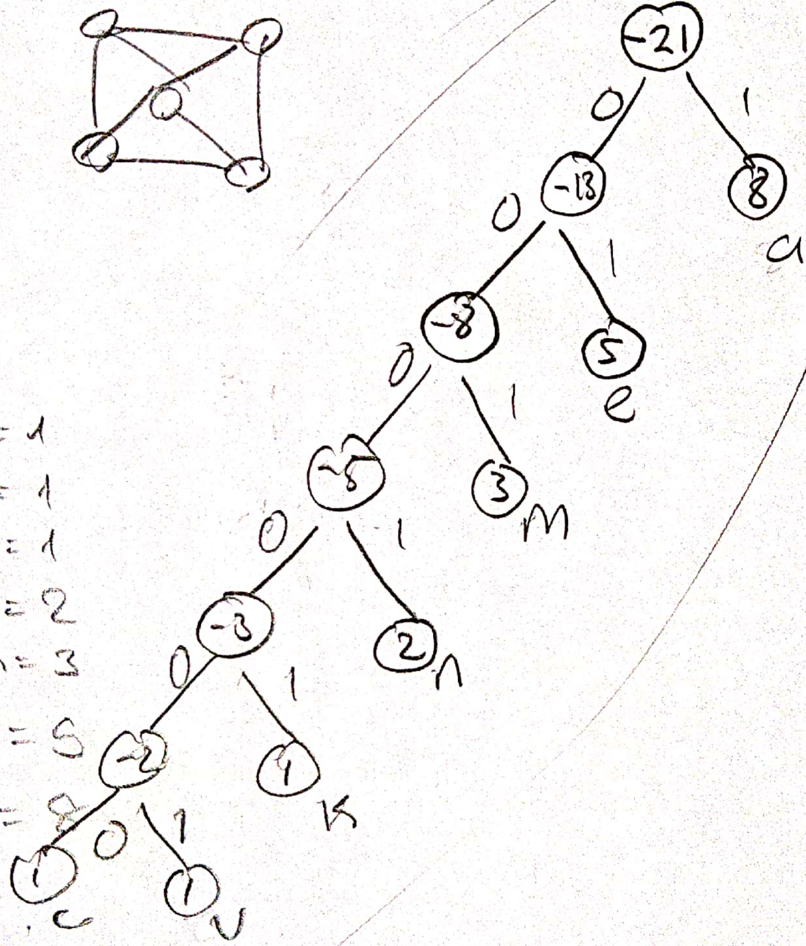
S_3: $c = a-b$

S_4: $w = c+1$

S_5: $d = a+2$

S_6: $w = w*d$





u = 1
c = 1
k = 1
n = 2
m = 3
e = 5
a = 8

a = 1 ⟶ 8
e = 01 ⟶ 10
m = 001 ⟶ 9
n = 0001 ⟶ 8
k = 00001 ⟶ 5
u = 000001 ⟶ 6
c = 000000 ⟶ 6

1 1 1 2 -2 3 -3 5 -5 8 -8 -13 -21

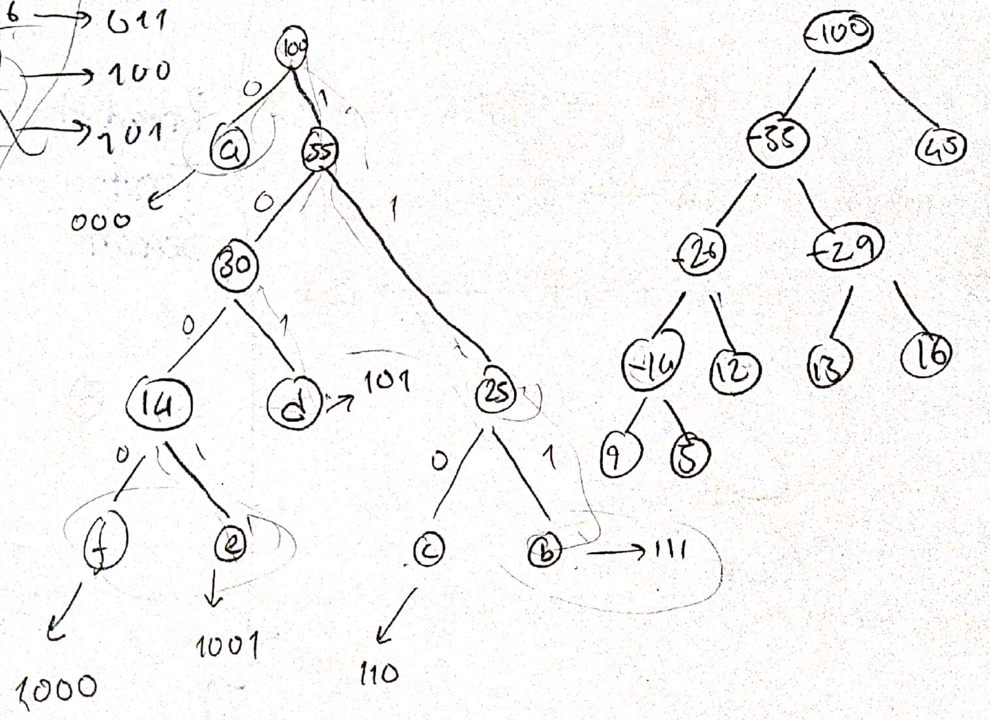
1 1 1 -2 2 -3 3 -5 5 -8 8 -13 -21

Huffman Trees

	frekans	
a →	45 →	000
b →	13 →	001
c →	12 →	010
d →	16 →	011
e →	9 →	100
f →	5 →	101

5 9 12 -14 13 16 -26 -29 45 -33

-100

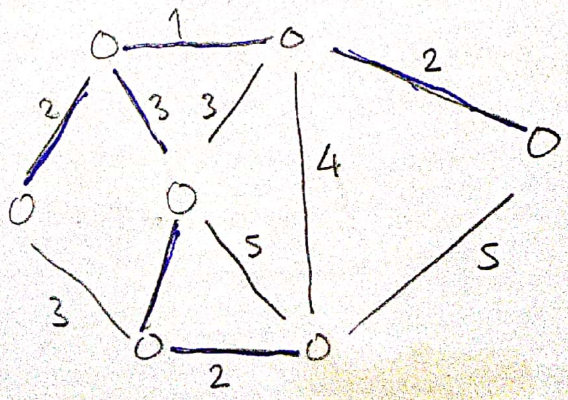


Kruskal's Algorithm

Adım 1 → En küçük ağırlığa sahip kenarı bul

Adım 2 → " " " " 2. kenarı bul ama kapalı döngü olmasın

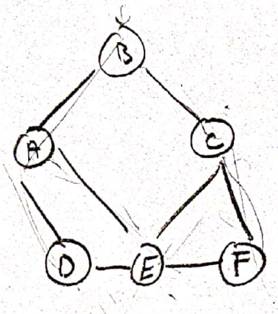
Adım 3 → Her düğümü dolaşana kadar adım 2'yi yap



Spanning Trees

G grafında T spanning tree

T G'nin subtreesi ve tüm nodeları içerir.



B → A/C → B ~~X~~

↳ A → D/E → A ~~X~~

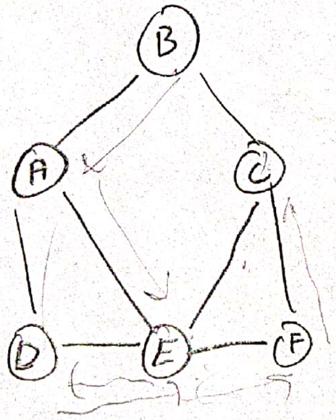
↳ C → F → C ~~X~~

↳ D

↳ E

↳ F

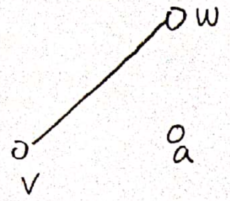
Breadth First Search



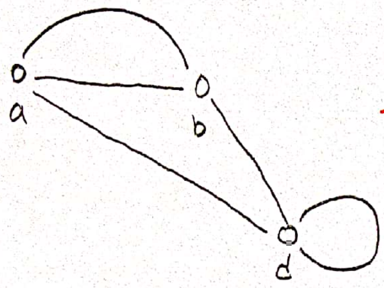
B → A → E → D → E → F → C

Depth First Search

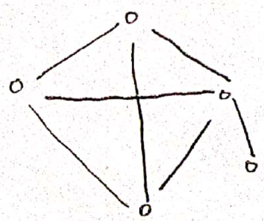
Graflar



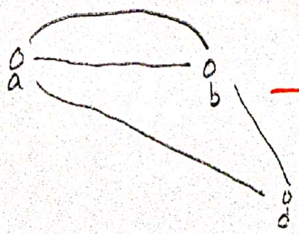
→ Ayrık düğüm (Isolated Vertex) = a



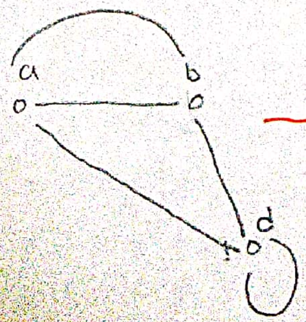
→
- a ve b iki paralel kenar ile birleştirilmiştir.
- d düğümünde döngü var



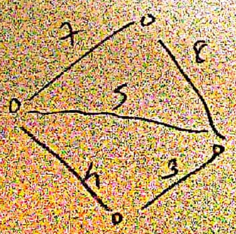
→ Basit (Simple) Graf yönsüz, paralelkenar olmayan ve döngü içermeyen graflardır.



→ Çoklu (Multi) graph, yönsüz, döngüsüz ama paralelkenar var.



→ Pseudo graph, yönsüz, döngülü, paralelkenar var.



→ Ağırlıklı (weighted) graf

Graflarda Benzerlik

$V(G) \longrightarrow \{v_1, v_2, v_3, v_4, v_5\}$

Her node v_i, bir üçlü ile gösterilir $\longrightarrow (p_1, p_2, p_3)$

$v_1 = (66, 20, 1)$

$v_2 = (41, 10, 2)$

$v_3 = (68, 5, 8)$

$v_4 = (90, 34, 5)$

$v_5 = (75, 12, 14)$

$v = (p_1, p_2, p_3)$, $w = (q_1, q_2, q_3)$

- $S(v,w) = \sum_{i=1}^{3} |p_i - q_i|$
- N seçilen sabit sayı seçilsin
- $s(v,w) < N$ ise v ve w arasına kenar eklenir.

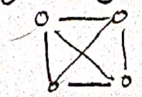
Eğer $v = w$ veya v w arasında yol varsa v w'nun aynı sınıfta olduğunu söyleyebiliriz.

Tam (Complete) Graf K_n

- $n \geq 3$
- K_n: n adet düğüm, her düğüm diğer düğümlerle bir kenar ile bağlanmıştır.

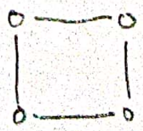
$n \rightarrow$ düğüm

$n \cdot (n-1)/2 \rightarrow$ kenar



Cycles (çember) Graf C_n

- $n \geq 3$
- $\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$ şeklindedir

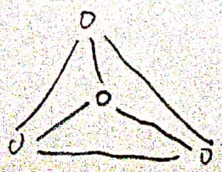
 $\longrightarrow C_4$

$n \rightarrow$ düğüm

$n \rightarrow$ kenar

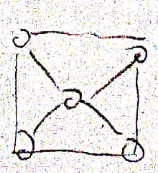
Wheel (Tekerlek) Graf W_n

- C_n grafına diğer bütün düğümlere bağlanacak şekilde düğüm eklenmesi.

 $\longrightarrow W_3$

$n+1 \longrightarrow$ düğüm

$2n \longrightarrow$ kenar

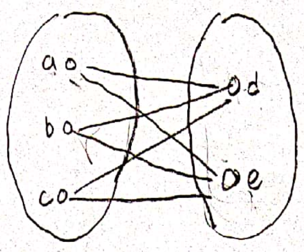


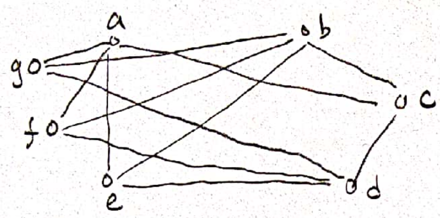
İki Parçalı (Bipartite) Graflar

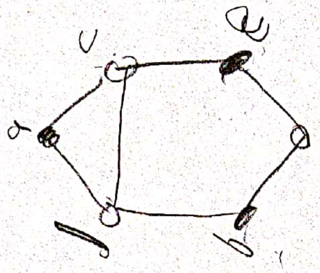
$V(G) = V(G_1) \cup V(G_2)$

$|V(G_1)| = m, \quad |V(G_2)| = n \longrightarrow$

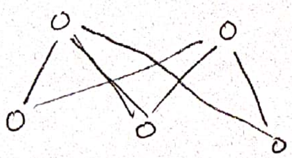
$V(G_1) \cap V(G_2) = \emptyset$



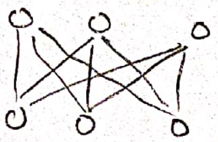




Tam (Complete) Bipartite Graf



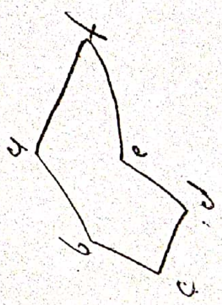
$K_{2,3}$



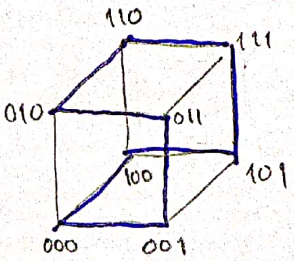
$K_{3,3}$

$m+n \longrightarrow$ düğüm

$m * n \longrightarrow$ Kenar

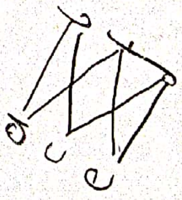


N-Cube Graf Q_n



$2^n \longrightarrow$ düğüm

$n \cdot 2^{n-1} \longrightarrow$ Kenar



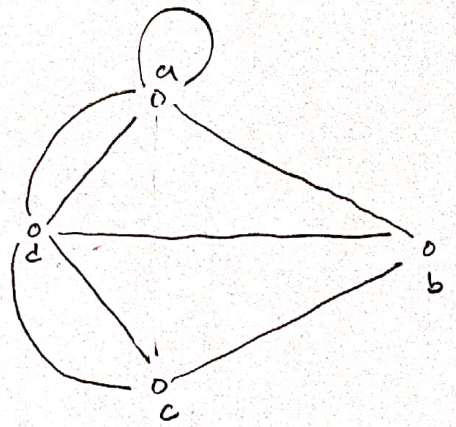
a b

c d

e f

Euler Döngüsü (Euler Cycles)

G grafı içerisindeki her kenardan 1 kez geçilerek başlangıç noktasına dönmek.



Düğüm Derecesi

$\delta(a) = 4$

$\delta(d) = 5$

Euler Grafı

G grafı Euler Cycle'ına sahipse Euler grafı denir

Tüm düğümlerin derecesi çift olmalı

Teorem

$e \longrightarrow$ kenar

$n \longrightarrow$ düğüm

$$\sum_{i=1}^{n} \delta(v_i) = 2e$$

$\delta(1) \rightarrow$ pendant

$\delta(0) \rightarrow$ isolated

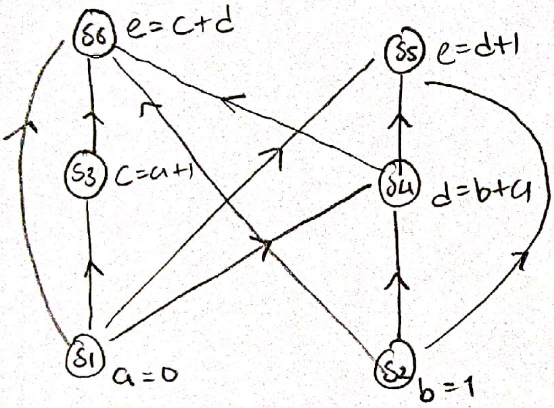
Hamiltonian Cycle

grafın üzerinde her node'dan 1 kez geçerek kapalı yol

G grafı sahipse hamiltonian graf

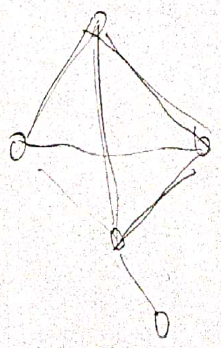
Precedence Graf

$S_1: a=0$ $S_2: b=1$ $S_3: c=a+1$ $S_4: d=b+a$ $S_5: e=d+1$ $S_6: e=c+d$

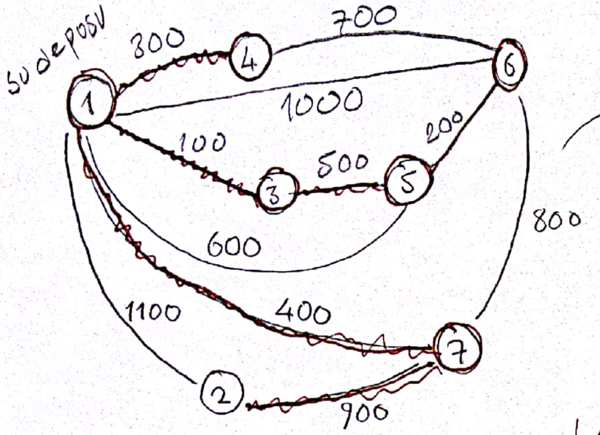


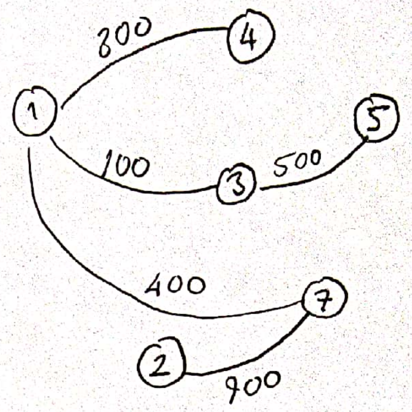
Planar Graf

Kenarlar birbirini kesmiyorsa planar.



Prim Algoritması (Minimum Spanning Tree)

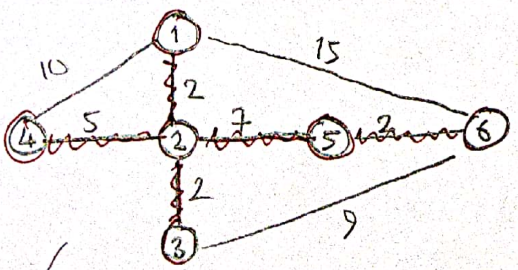


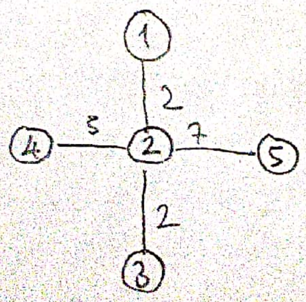


		1	2	3	4	5	6	7	
1- X	1	m	1100	100	300	600	1000	400	800
7-X	2	1100	m	m	m	m	m	900	
2- X	3	100	m	m	m	$\boxed{500}$	m	m	500
3- X	4	300	m	m	m	m	700	m	
5-X	5	600	m	500	m	m	$\boxed{200}$	m	200
6-X	6	1000	m	m	700	200	m	800	
4-X	7	400	$\boxed{900}$	m	m	m	800	m	900

\+

2400 br

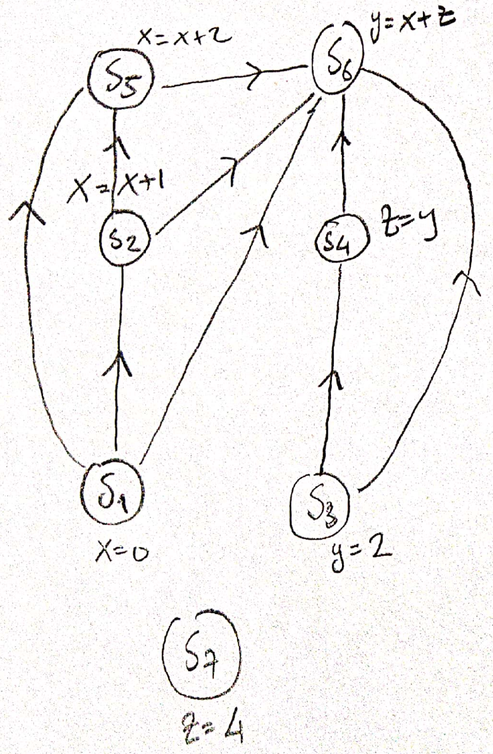




		1	2	3	4	5	6	
X	1	m	2	m	10	m	15	2
X	2	2	m	2	5	7	m	14
X	3	m	2	m	m	m	9	
X	4	10	5	m	m	m	m	
X	5	m	7	m	m	m	2	2
X	6	15	m	9	m	2	m	

18 br

$S_1: X=0 \quad S_2: X=X+1 \quad S_3: y=2 \quad S_4: z=y \quad S_5: X=X+2 \quad S_6: y=z+X \quad S_7: z=4$

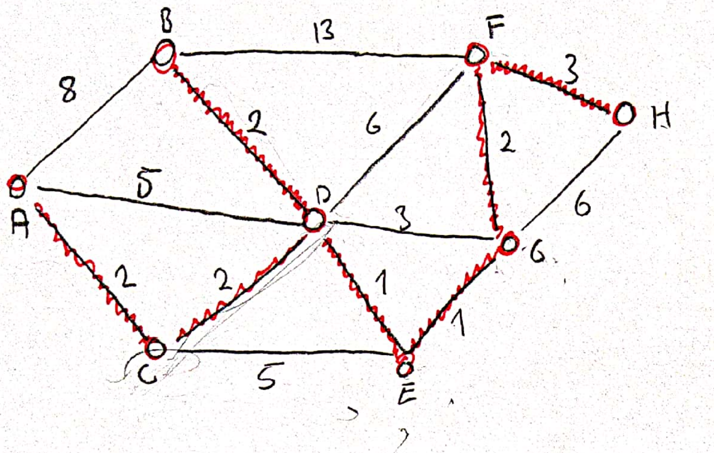


geçmişe bak

precedence graph

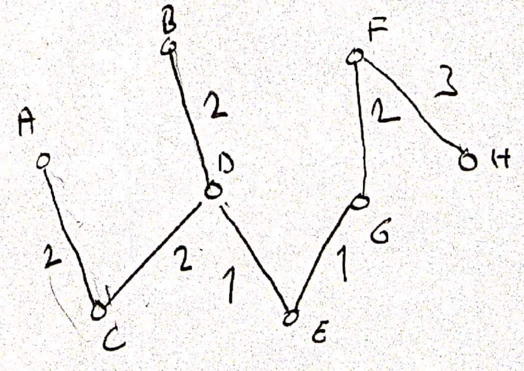
29.11.2019

Dijkstra Algoritması



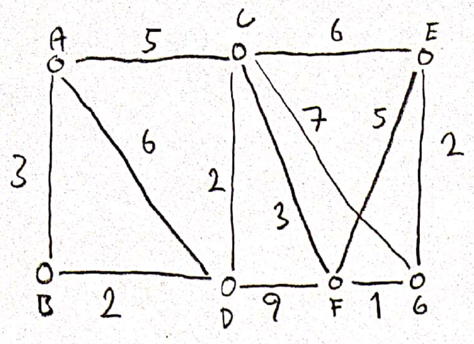
A

	A	B	C	D	E	F	G	H
A	$\boxed{0_A}$	8_A	2_A^*	5_A	∞	∞	∞	∞
C		8_A	$\boxed{2_A}$	4_C^*	7_C	∞	∞	∞
D		6_D		$\boxed{4_C}$	5_D^*	10_D	7_D	∞
E		6_D^*			$\boxed{5_D}$	10_D	6_E	∞
B		$\boxed{6_D}$				10_D	6_E^*	∞
G						8_G^*	$\boxed{6_E}$	12_G
F						$\boxed{8_G}$		11_F^*
H								$\boxed{11_F}$



	A	B	C	D	E	F	G	H
A	$\boxed{0_A}$	8_A	2_A^*	5_A	∞	∞	∞	∞
C		8_A	$\boxed{2_A}$	4_C^*	7_C	∞	∞	∞
D		6_D		$\boxed{4_C}$	5_D^*	10_D	7_D	∞
E		6_D^*			$\boxed{5_D}$	10_D	6_E	∞
B		$\boxed{6_D}$				10_D	6_E^*	∞
G						8_G^*	$\boxed{6_E}$	12_G
F						$\boxed{8_G}$		11_F^*
H								$\boxed{11_F}$

Kruskal Alg...



	A	B	C	D	E	F	G
A	$\boxed{A_0}$	3_A^*	5_A	6_A	∞	∞	∞
B		$\boxed{3_A}$	5_A	5_B^*	∞	∞	∞
D			5_A^*	$\boxed{5_B}$	∞	14_D	∞
C			$\boxed{5_A}$		11_C	8_C^*	12_C
F					11_C	$\boxed{8_C}$	9_F^*
G							$\boxed{9_F}$

Finite State Automata

Initial State (S_0)

Durumlar listesi (S)

Her durum geçişi için olması gereken koşullar listesi (conditions) f, g...

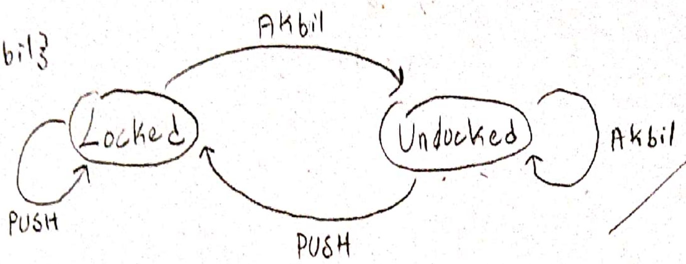
- I, O

FSM'nin hafızası : State #

Örnek:

$S = \{Locked, Unlocked\}$

$I : \{Push, akbil\}$



turnike

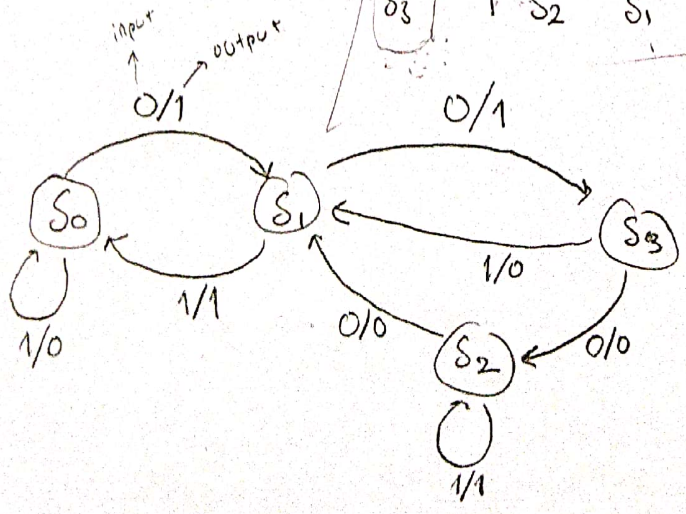
Örnek:

$S = \{S_0, S_1, S_2, S_3\}$

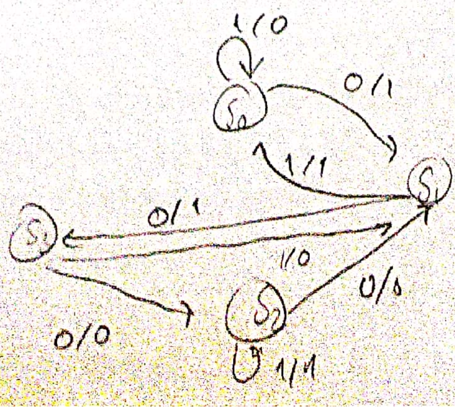
$I = \{0, 1\}$

$O = \{0, 1\}$

State	Input 0	Input 1	Output	
S_0	S_1	S_0	1	0
S_1	S_3	S_0	1	1
S_2	S_1	S_2	0	1
S_3	S_2	S_1	0	0



rand() % 4



60% 30 / 100

18 + 10,5

28,5

12

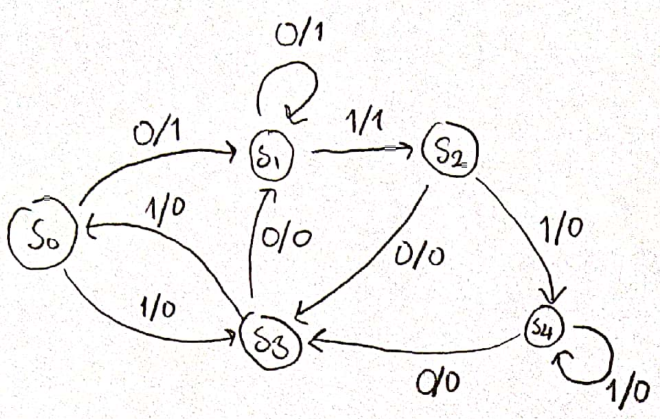
68,5

35.30 / 100 105

3
+2.100 / 40

100.40 / 100 40

örnek:



$I = \{0, 1\}$

$O = \{0, 1\}$

$S = \{s_0, s_1, s_2, s_3, s_4\}$

State	Input 0	Input 1	Output	
s_0	s_1	s_3	1	0
s_1	s_1	s_2	1	1
s_2	s_3	s_4	0	0
s_3	s_1	s_0	0	0
s_4	s_3	s_4	0	0

örnek:

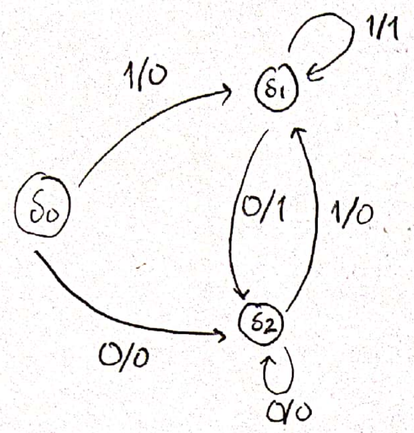
Yukarıdakine göre

Input: 101011

$\longrightarrow$ Output: 001000

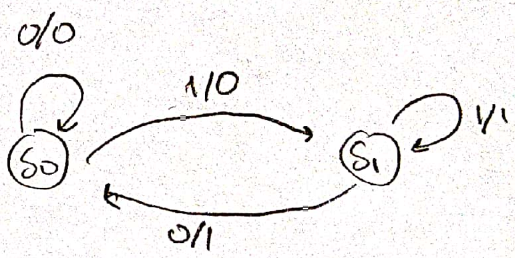
Outputs?

örnek:



Inputs	Output
10101	01010
1111	0111
0000	0000
010100	001010

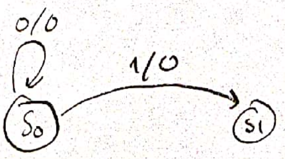
Sağa 1 bit shift



örnek

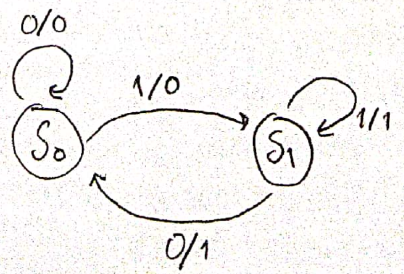
Sola shift eden FSM

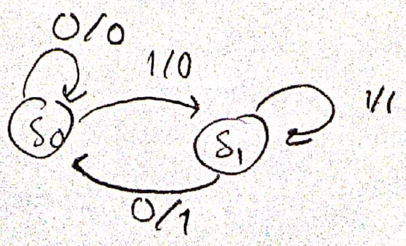
Input	Output
0110	1100
1100	1000
0001	0010



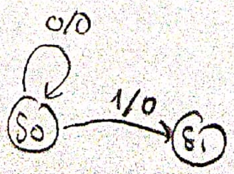
0110
1100

0110
1100





1100 1000

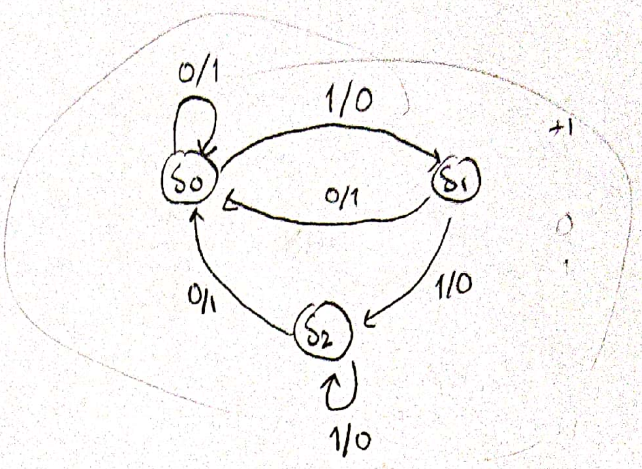


0

1111
1110
1010

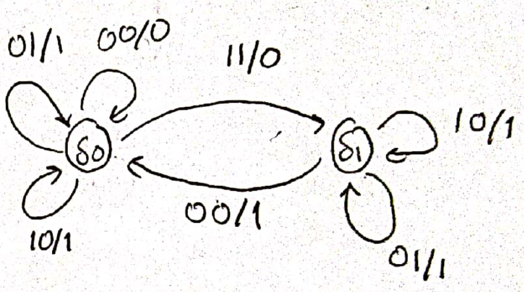
örnek

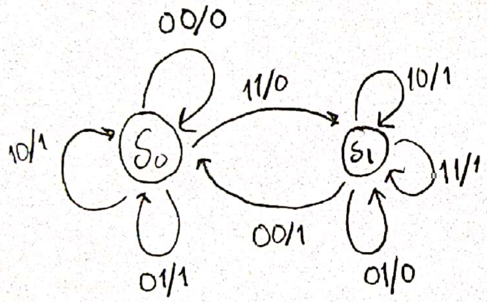
Binary toplama



I_1	I_2	Output
0	0	0
0	1	1
1	0	1
1	1	0

Eg = 1





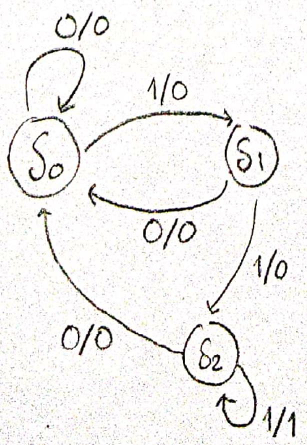
  00111
+00110
01101

örnek

gönderilen inputta art arda 3 tane 1 varsa hata mesajı gönderilsin.

I: 1 1 1 1 0 0 1 1 0 1 0 0 1 1 1

O: 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1

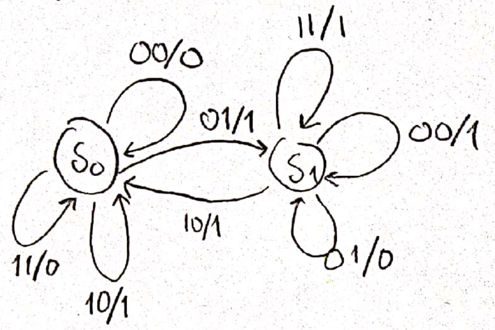


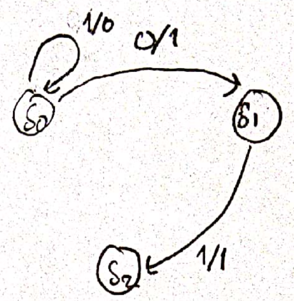
Örnek

Çıkarma 2 bit

I	O
0 0	0
0 1	1 (borç)
1 0	1
1 1	0

```
    1101 0 1
  - 0110 0 1
  ----------
 2  0111 0 0
```





Örnek

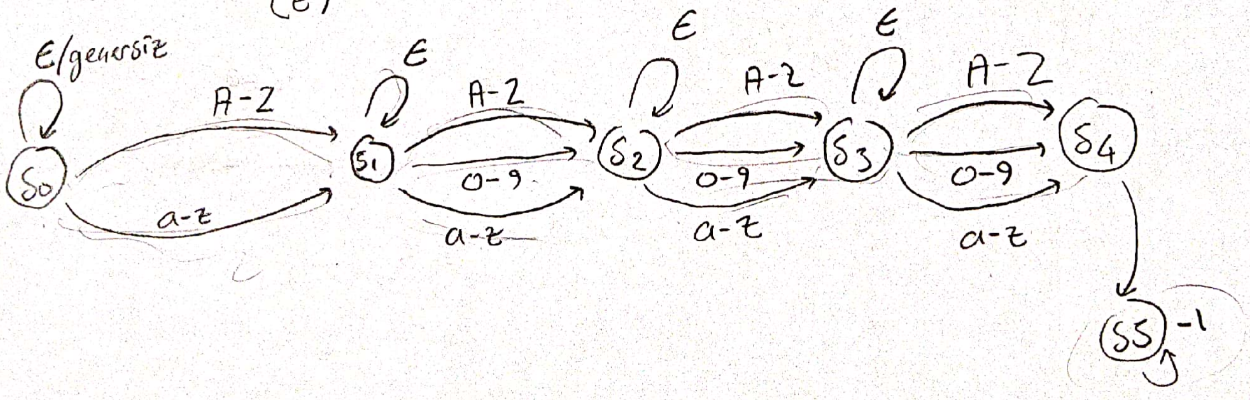
Değişken ismi 4 char uzunluğunda

ilk karakter $A < - < Z$ aralığında olmalı

Sonrası
a-z
A-Z
0-9

Başka karakter girilirse geçersiz sayılacak ve yeni bir karakter isteyecek

(ε)



Örnek

$\boxed{\text{login}}$

kullanıcı ID'si isteniyor

ID geçersizse yeniden ID

ID geçerliyse password

password yanlış ise ID girişi tekrarlanır

" doğru " ✓

v → valid ID

i → invalid ID

p → valid password

q → invalid password

a → "enter ID"

b → "enter password"

c → prompt

x → any input

qt → quit