BİLGİSAYAR BİLİMLERİNDE GÜNCEL KONULAR II

Hafta 7

. Ağaç Graflar

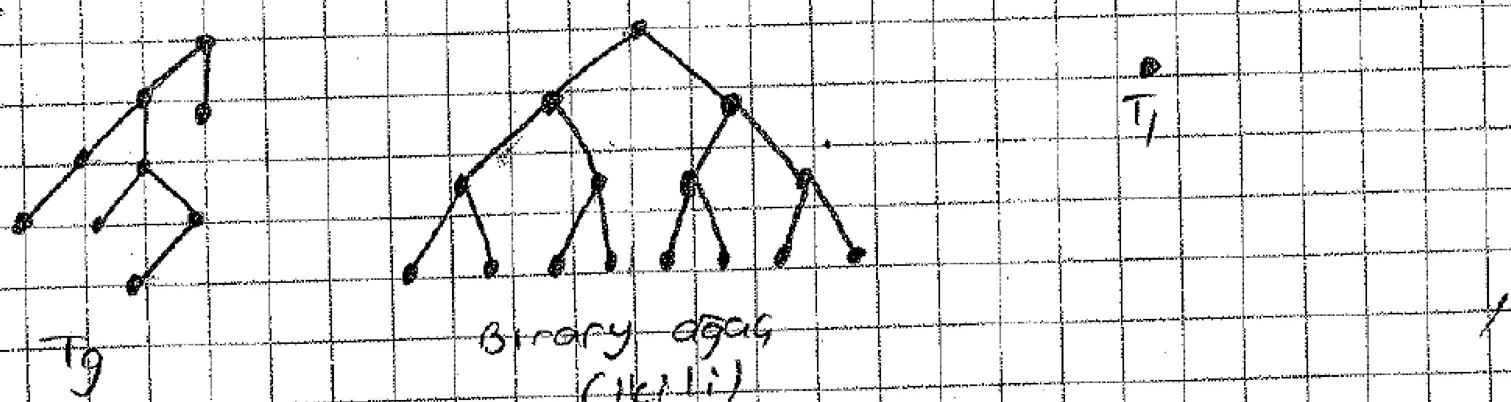
. Ağaç Grafların Bilgisayarlarda Saklanması

. Dallanmış Alt ağaçlar

Ağaç Graflar

AĞAÇ GRAF

Çevre içermeyen birleştirilmiş bir grafa ağaç adı verilir. n tepeli bir ağaç genellikle T_n ile gösterilir.



⇒ Önemli ağaç graf arasına yol graf ve yıldız grafta girer.

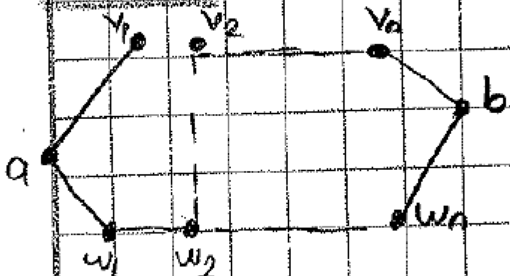
Teorem

Bir T ağacının herhangi 2 tepe a ve b olsun. Bu iki tepe arasında 1 tek yol vardır.

İspat

Olmayana ergi ile 2 tane yol old. kabul edelim.

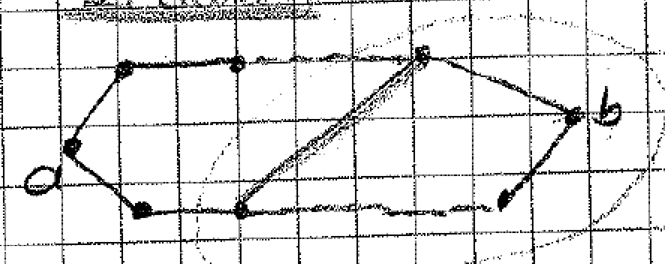
1. Durum



$v_2 w_2 \ldots b w_n$ şeklinde bir çevre graf oluşur.
Ağaç graf olmaz.

v_1 ile v_2 arasındaki ayrıtı sildik yerine w_2 ekledik.

2. durum



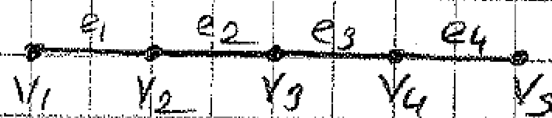
Çevre oluştu. Çelişki.

herhangi bir ayrıt ekledik.

[illegible]

p tepeli q ayrıtlı bir ağaç grafta $p = q + 1$ dir. Kanıtlayınız.

P_5



$$\left.\begin{array}{l} p = 5 \\ q = 4 \end{array}\right\} \Rightarrow p = q + 1$$

Kanıtını $\left.\begin{array}{l} p = 2 \text{ için} \\ k = n \\ k+1 = n \end{array}\right\}$ tümevarımla ispatlayın.

Ağaç Grafların Bilgisayarlarda Saklanması

Tanım: Çevre içermeyen bağlantılı bir grafa ağaç denir.

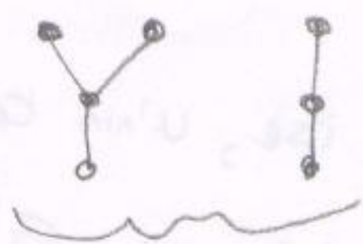
Örn)

G_1 G_2 G_3 G_4

$\Rightarrow$ Hepsi ağaçtır.

Tanım: Birden fazla ağaç grafın birleşmesiyle oluşan grafa orman denir.

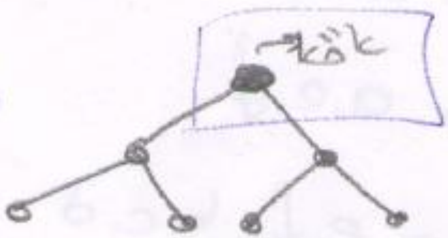
Örn)



G

$\Rightarrow$ G, 2 parçalı bir ormandır.

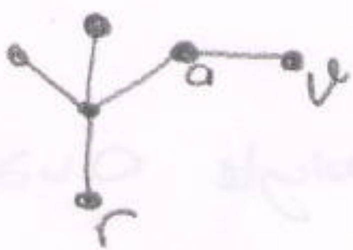
Tanımı: Çoğu zaman tepelerinden biri özel olan ağaçlar ile çalışılır. Özel olarak seçilen bu tepeye kök denir. Bu grafa da köklü ağaç denir.



köklü ağaç

Tanım: G köklü bir ağaç ve r noktası da G'nin kökü olsun. G ağacının r noktasından farklı olan bir v noktasını alalım. v ile r noktasını birleştiren yolda v noktasının komşusu olan noktaya v noktasının babası denir.

(n,m)



v'nin babası a'dır.

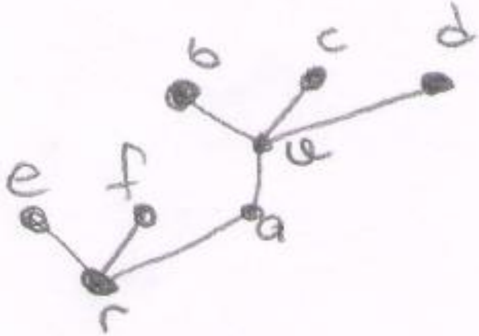
v'nin babası r'dir.

Tanım: u noktasının diğer noktalarına ise, u'nin çocukları denir. G çizgesinin çocuğu olmayan noktalarına G'nin yaprakları denir.

1 dereceli tepelere G'nin yaprakları denir.

Uyarı: r bir kök olmak üzere r'nin babası yoktur.

(Çizim)



G grafı

u'nin babası = a

çocukları = b, c, d

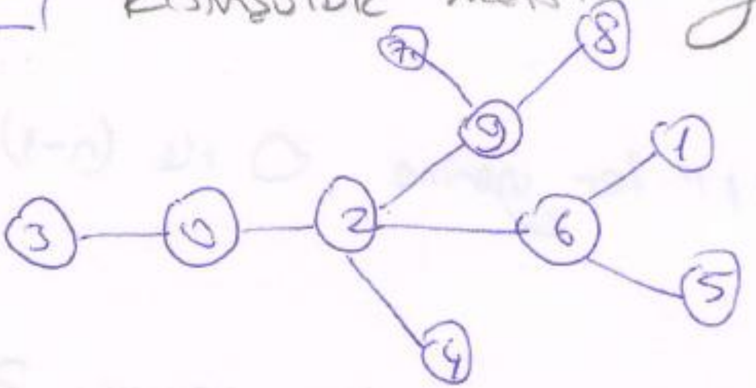
r'nin çocukları = a, e, f

G'nin yaprakları = e, f, b, c, d

= Ağaçları Bilgisayar Belleğinde Nasıl saklarız? =

4 farklı yöntem ile ağaçları bilgisayar belleğinde saklayabiliriz.

1. yöntem: Komşuluk matrisi yardımıyla.



G

G ağacını aşağıdaki matris yardımıyla bellekte saklayabiliriz.

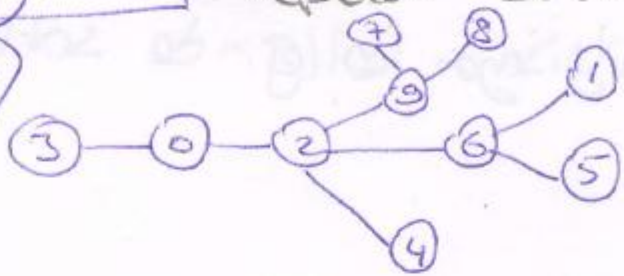
G ağacını aşağıdaki matris yardımıyla bellekte saklayabiliriz.

	1	2	3	4	5	6	7	8	9	0
1	0	0	0	0	0	1	0	0	0	0
2	0	0	0	1	0	1	0	0	1	1
3	0	0	0	0	0	0	0	0	0	1
4	0	1	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	1	1	0	0	1	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	0	1	0
9	0	1	0	0	0	0	1	1	0	0
0	0	1	1	0	0	0	0	0	0	0

(*) Bu yöntemle; n tepeli bir ağacı saklamak için n^2 bit gereklidir. Fakat matris simetrik olduğundan ve köşegen elemanları sıfır olduğundan $\frac{n^2-n}{2}$ bit yeterlidir.

2. yöntem / Tepeleri saklamayıp, sadece kenarları saklarız.

a) n



7	8	9	6	3	0	2	6	6
9	9	2	2	0	2	4	1	5

Böylece, bu yöntemde sadece 2 satır kullandık ve (n−1) sütun kullanıldı.

Ancak bu kez, "0" ile "1" ler yerine 0 ile (n−1) arasındaki tamsayıları kullandık.

n pozitif bir tamsayı olmak üzere bu sayıyı 2'lik sistemde yazmak için $\log_2 n + 1$ bit gerekir.

Bu durumda; n tepeli bir ağacı bellekte saklamak için;

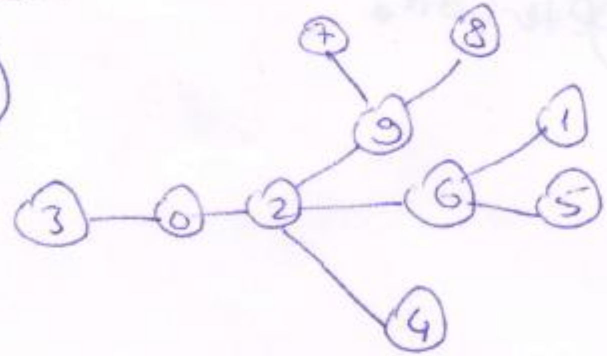
$$2\cdot(n-1)\cdot(\log_2 n+1) < \frac{n^2-n}{2}$$

Böylece, 2. yöntem için daha az bit gereklidir.

3. yöntem (Father Code)

Bu yöntemde 0 nolu tepeyi kök kabul ederiz. 2. yönteme benzer olarak yine kenarlar saklanır. Ancak kural üst satıra çocuklar, alt satıra babalar yazılır.

Ör.)



çocuklar	1	2	3	4	5	6	7	8	9
babalar	6	0	0	2	6	2	9	9	2

Bu yöntemde,

→ O (kök) üst satırda bulunmaz.

→ O harica tüm noktalar üst satırda bulunmak zorundadır.

→ Üst satırda hiç bir rakam 2 kez kullanılmaz.

→ Üst satırdaki sayıları sıralamaya gerek yoktur. Çünkü

sıralılar.

→ Sadece alt satır sıralanır.

→ Sonra olarak sıralamak için $(n-1).(\log_2 n+1)$ bitle ihtiyaç

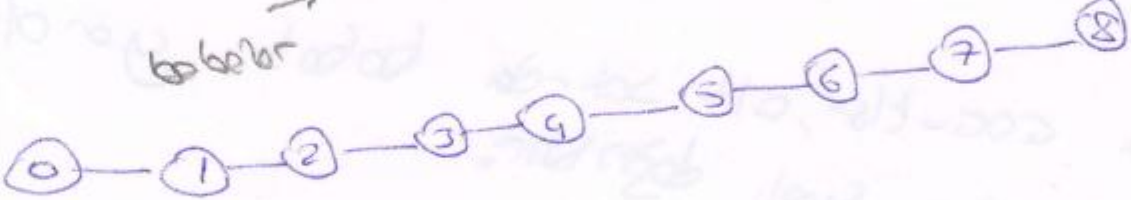
vardır.

★ Her ağaç için bu kod yazılabilir. Fakat her kod

bir ağaç belirtmeyebilir!!

4m) $(0,1,2,3,4,5,6,7)$ dizisi bir ağaç için "father code" olurmu? $\Rightarrow$ 8 ayrıt varsa 9 tepe vardır.

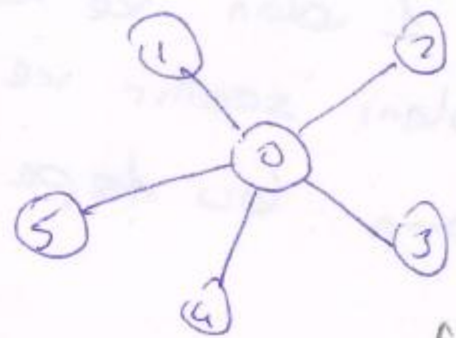
çocuklar $\rightarrow$	1	2	3	4	5	6	7	8
babalar $\rightarrow$	0	1	2	3	4	5	6	7



$\Rightarrow P_9$ grafı

4n) $(0,0,0,0,0)$ dizisi bir ağaç için "father code" olurmu? $\Rightarrow$ 5 ayrıt varsa 6 tepe vardır.

çocuklar $\rightarrow$	1	2	3	4	5
babalar $\rightarrow$	0	0	0	0	0



$K_{1,5}$ yıldız grafı.

m)

(7, 6, 5, 4, 3, 2, 1, 0) bir ağacın "Prüfer code" olur mu?

1	2	3	4	5	6	7	8
7	6	5	4	3	2	1	0

→ Prüfer code olamaz.

m)

(2, 3, 1, 2, 3, 1, 2, 3) "bir ağacın "Prüfer code" olur mu?

1	2	3	4	5	6	7	8
2	3	1	2	3	1	2	3

⇒ 0 (kök) yoktur.

Prüfer code olamaz!!

4. yöntem (Prüfer Kod)

Bu yöntem 3. yöntemin iyileştirilmesidir. O yine kök olarak kabul edilir.

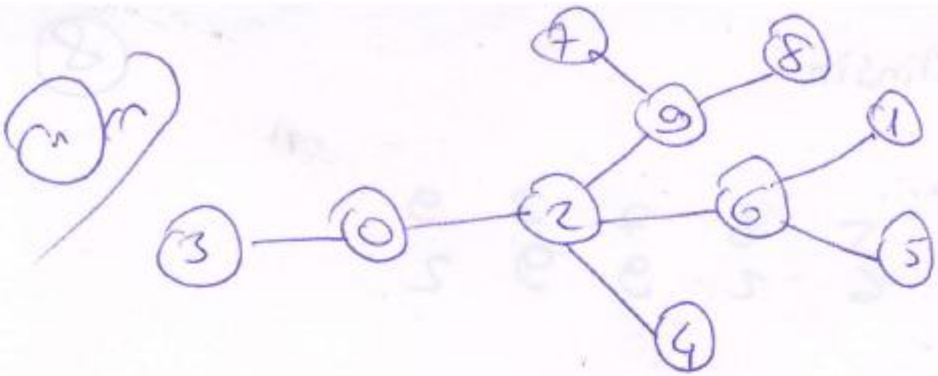
Bu yöntemde n tepeli bir ağaç için (n-1) ayrıt yerine (n-2) ayrıtı saklarız.

yöntemde;
üst satırda çocuklar, alt satırda babalar vardır.

Ancak bu sefer üst satır sıralı değildir.
Sıralama şu şekilde yapılır;

A1: Derecesi 1 olan ve kökten farklı olan noktalardan en küçük olanı seçilir ve babasıyla birlikte yazılır.

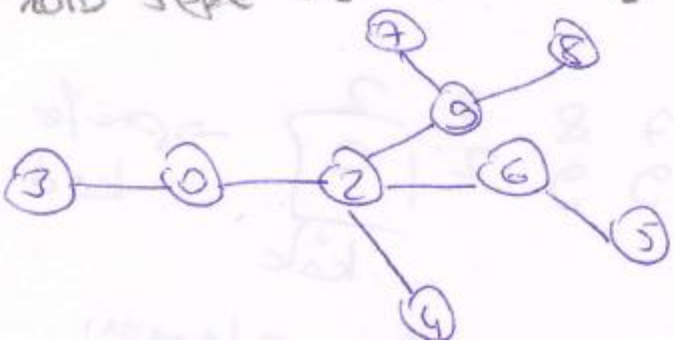
A2: Daha sonra bu tepe ve bitişik ayrıtı silerek A1'e git.



1 nolu tepeyi seçelim

1
6

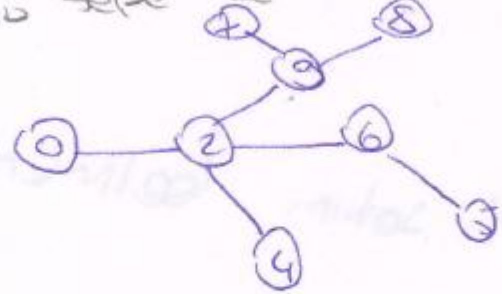
1 nolu tepe ve bitişik ayrıtı silindi.



Şimdi 3'ü seçelim.

1 3
6 0

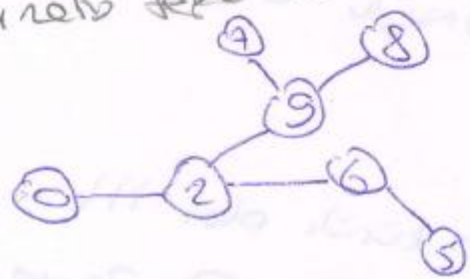
3 nolu tepe ve bitişik ayrıtı silindi.



Şimdi 4'ü seçelim.

1	3	4
6	0	2

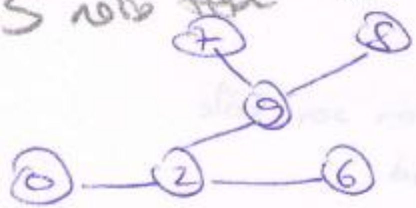
4 nolu tepe ve bitişik ayrıt silindi.



Şimdi 5'i seçelim.

1	3	4	5
6	0	2	6

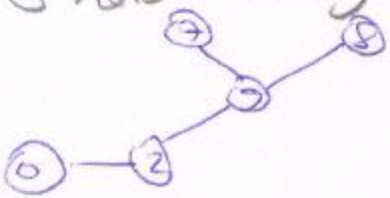
5 nolu tepe ve bitişik ayrıt silindi.



Şimdi 6'yı seçelim.

1	3	4	5	6
6	0	2	6	2

6 nolu tepeyi ve bitişik ayrıtı silelim.



Şimdi 7'yi seçelim.

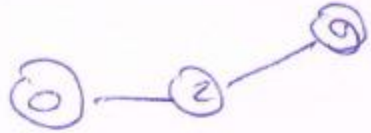
1	3	4	5	6	7
6	0	2	6	2	9

7 ve bitişik ayrıt silindi

8'i seçelim

1	3	4	5	6	7	8
6	0	2	6	2	9	9

8 nolu tepe ve bitişik ayrıt silinsin.



9'u seçelim.

1	3	4	5	6	7	8	9
6	0	2	6	2	9	9	2

9 nolu tepe ve bitişik ayrıt silinsin.

2'yi seçelim.

1	3	4	5	6	7	8	9	2
6	0	2	6	2	9	9	2	$\boxed{0}$

→ prüfer kod

kök

prüfer kod: (6, 0, 2, 6, 2, 9, 9, 2)

Extended prüfer kod = (6, 0, 2, 6, 2, 9, 9, 2, 0)

En son kök olacağından alt satırın son elemanı 0(sıfır) olur.

Prüfer dizisinin 2. satırı 1. satırı belirler.

Teorem: Prüfer dizisinin 2. satırı 1. satırı belirler.

(n)m

Derece dizisi (3, 2, 1, 2, 1, 1) ve prüfer codu'u (0, 1, 0, 3) olan ağacı çiziniz.

Tepe	0	1	2	3	4	5
Derece	3	2	1	2	1	1

(*) Derecesi en küçük olan ilk tepe 2 nolu tepedir. O zaman

2 ? ? ? ?
0 1 0 3 0 → en son kök

extended prüfer kod.

Şimdi 2'yi 0'ın derecesini 1 indirelim.
Sabit oldu dan 2 ile Şimdi 1 olmalı.

Tepe	0	1	2	3	4	5
Derece	2	2	0	2	1	1

(*) Derecesi en küçük ilk tepe 4
O zaman

2 4 ? ? ?
0 1 0 3 0

4'ün ve 1'in derecesini 1 azaltalım.

En küçük dereceli ilk tepe = 1

Tepe	0	1	2	3	4	5
Derece	2	1	0	2	0	1

2 4 1 ? ?
0 1 0 3 0

Şimdi 0'ın ve 1'in derecesini 1 azaltalım.

0 kalır (kök) en küçük dereceli tepe = 5

Tepe	0	1	2	3	4	5
Derece	1	0	0	2	0	1

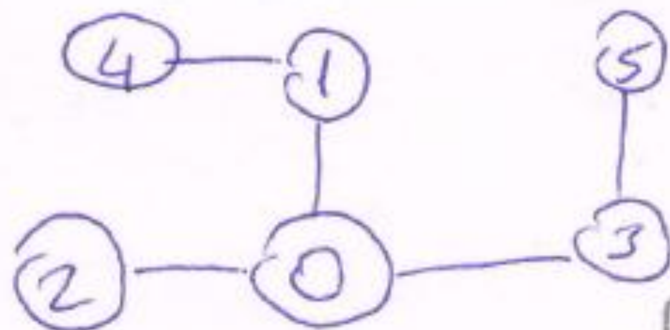
2 4 1 5 ?
0 1 0 3 0

Şimdi 5 ve 3'ün derecesi 1 azalsın.

0 kalır derecesi en küçük tepe = 3

Tepe	0	1	2	3	4	5
Derece	1	0	0	1	0	0

2 4 1 5 3
0 1 0 3 0



G gráf.

Tanım: (Düzlemsel Kod):

Tepe sayısı n olan bir ağacı alalım. Bu ağacın kökü r olsun ve r'den başlayıp ağacın tüm ayrıtlarından 2 kez geçerek bir tur yapalım. Bu turda ulaştığımız tepe bir önceki tepenin çocuğu ise bunu "1" ile, babası ise "0" ile gösterilmek üzere elde edilen koda düzlemsel kod denir.

Örnek

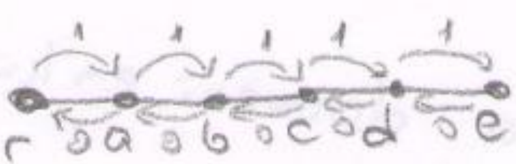
Turumuz

r e f g f e c a b a b c d c e r

1 1 1 0 0 1 1 1 0 0 1 0 0 0

Örn

111100000 düzlemsel koduna karşılık gelen ağaç.



= P6 yol grafi

(1.) m) 101010 düzlemsel koduna karşılık gelen ağaç.

rarbrcr
vvvvvv
101010 $= K_{1,3}$ yıldız grafı

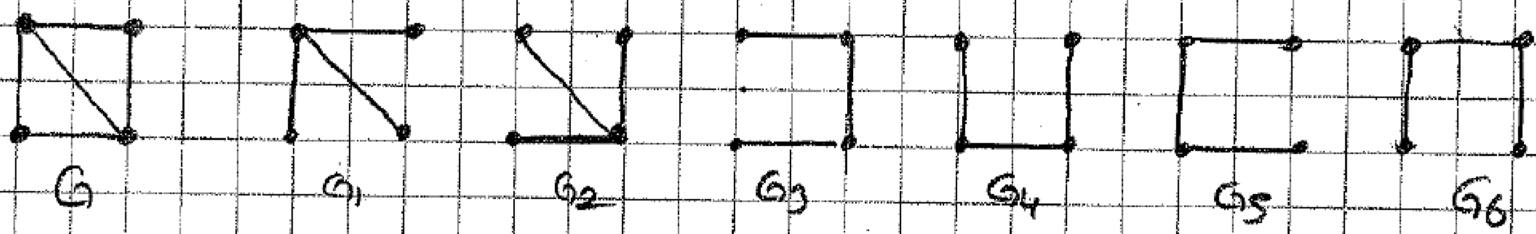
(1.) m) 1100011100 düzlemsel kodlu ağacı çiziniz.

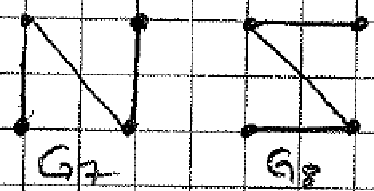
En başta 2 tane 1, fakat sonra arka arkaya 3 sıfır olduğundan böyle bir ağaç çizilemez.

Dallanmış Alt ağaçlar

Bir G grafının tüm tepelerini içeren birleştirilmiş bir alt grafa dallanmış alt graf denir. Eğer dallanmış alt graf çevre içermiyorsa dallanmış (alt) ağaç denir.

ÖRN:





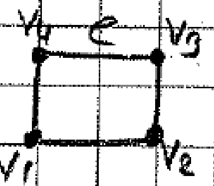
NOT ⇒ Tüm tepeleri içerecek.

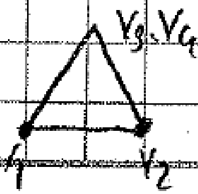
Teorem: (Cayley)

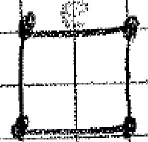
$Z(G)$, dallanmış ağaçların sayısını göstermek üzere,

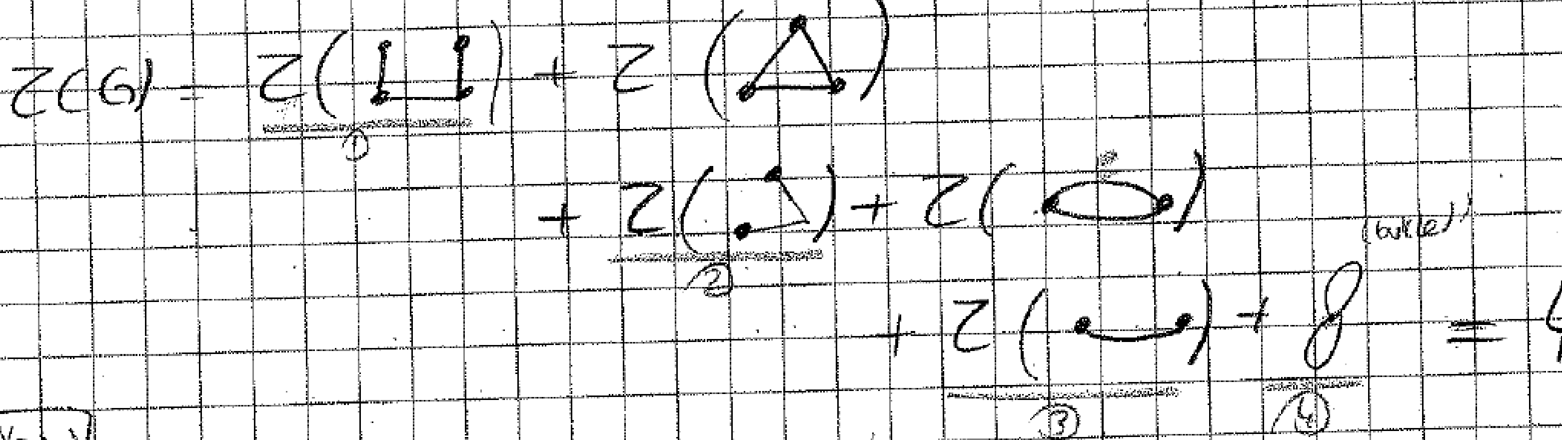
$$Z(G) = Z(G-e) + Z(G.e)$$ büzülme

⇒ **büzülme işlemi:**

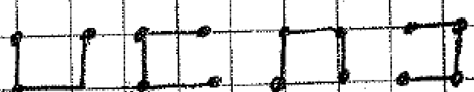


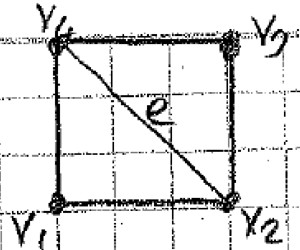
$G.e =$ 

 = 4



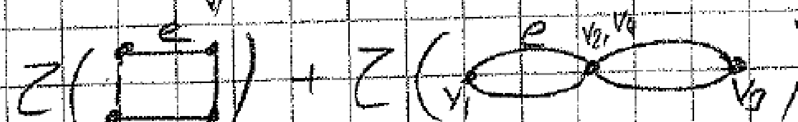
Yön 1

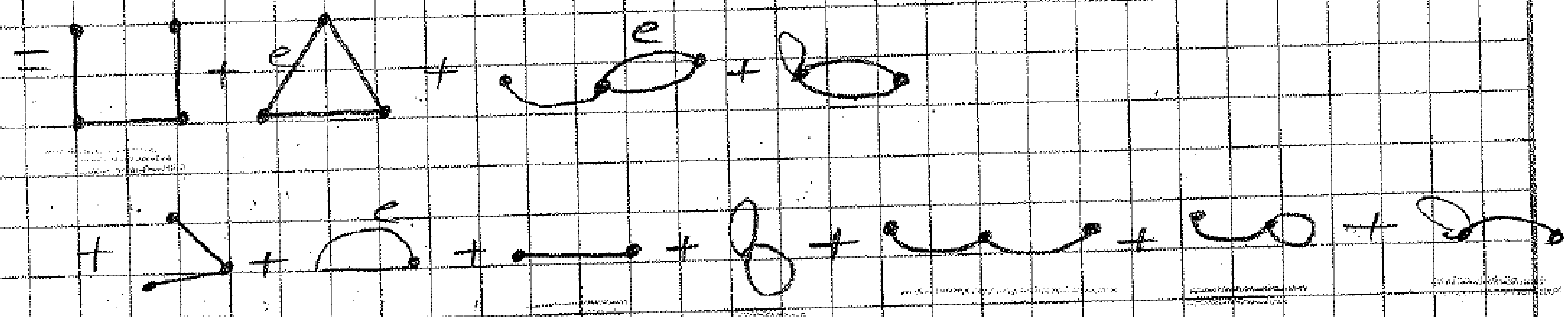
 ⇒ 4 tane

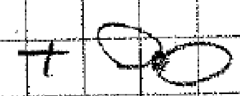


G

$$\tau(G) = \tau(G-e) + \tau(G.e)$$







$= 8$

Bir G grafının Dallanmış alt ağaç sayısı kaçtır?

KAYNAKLAR

[1] Chartrand, G.-Lesniak, L., (1986) : *Graphs and Digraphs*, Wadsworth & Brooks, California

[2] West D.B. (2001) : *Introduction to Graph Theory, Prentice Hall*, USA.

[3] Graf Teoriye Giriş, Şerife Büyükköse ve Gülistan Kaya Gök, Nobel Yayıncılık

[4] Discrete Mathematical Structures for Computer Science, Ronald E. Prather, Houghton Mifflin Company, (1976).

[5] Christofides, N., 1986. Graph Theory an Algorithmic Approach, Academic Press, London