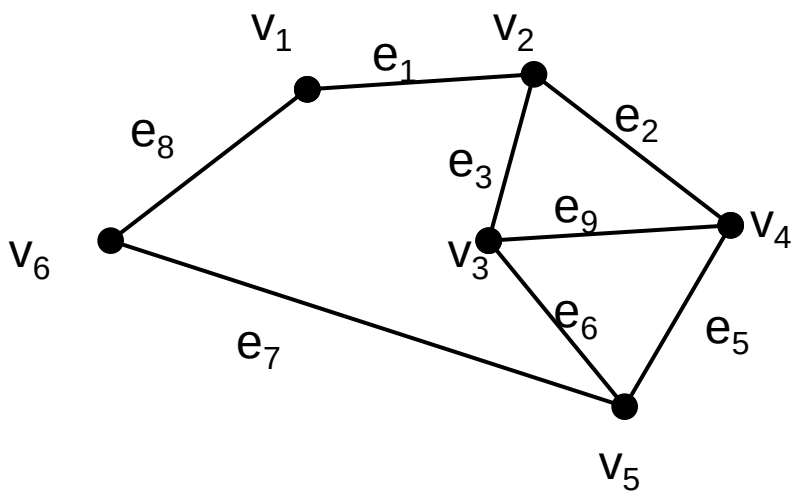
BİLGİSAYAR BİLİMLERİNDE GÜNCEL KONULAR II

Hafta 2

Alt graf :G=(V,E) bir graf olsun. $V' \subseteq V$, $E' \subseteq E$ olmak üzere $G'=(V',E')$ grafına G nin bir altgrafı denir.



G

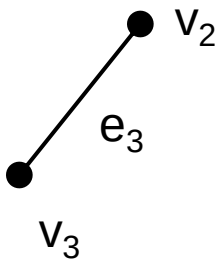
Alt graf örneği(1)

G'

$V'=\{v_1\}$

$E'=\varnothing$

Alt graf örneği(2):



G'

$V'=\{v_2,v_3\}$

$E'=\{e_3\}$

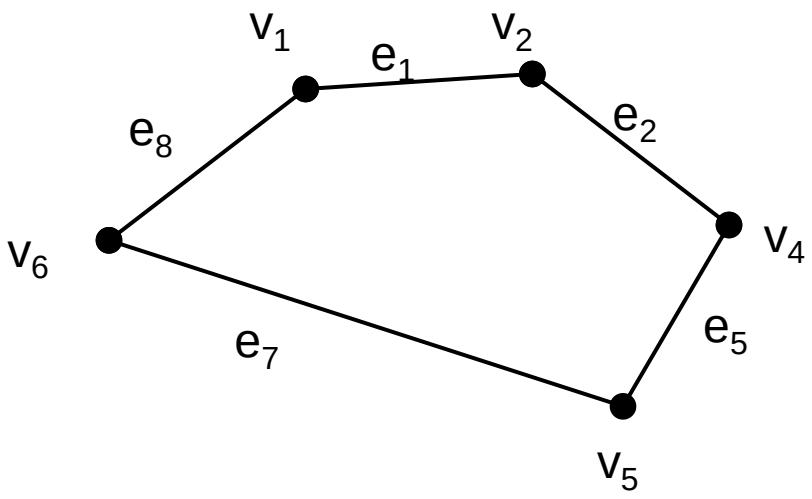
Alt graf örneği(3):

G'

$V'=\{v_1,v_4\}$

$E'=\varnothing$

Alt graf örneği(4):

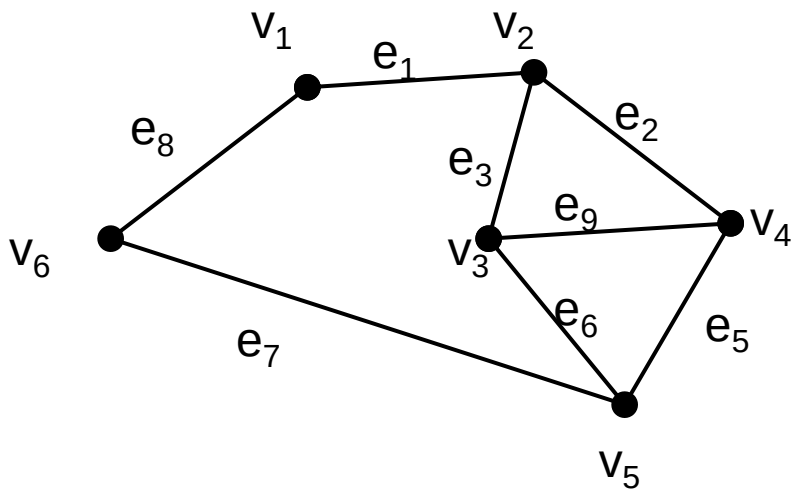


G'

$V'=\{v_1,v_2,v_4,v_5,v_6\}$

$E' =\{e_1,e_2,e_5,e_7,e_8\}$

Etkilenmiş Alt Graf : G=(V,E) bir graf olsun. V'⊆V alt kümelerindeki tepeler ile, bu tepeler arasında G de bulunan ayrıtların oluşturduğu grafa etkilenmiş alt graf denir.

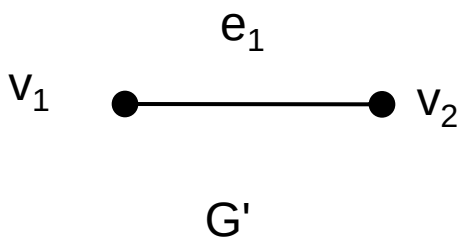


G

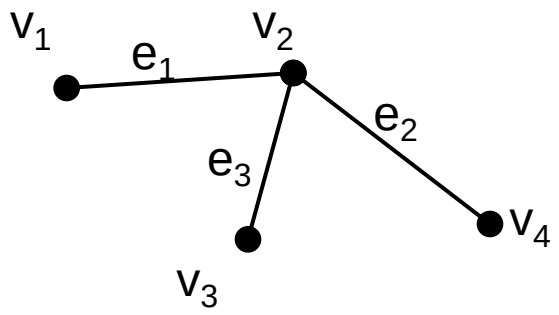
Etkilenmiş alt graf örneği(1):

G'

Etkilenmiş Alt graf örneği(2):

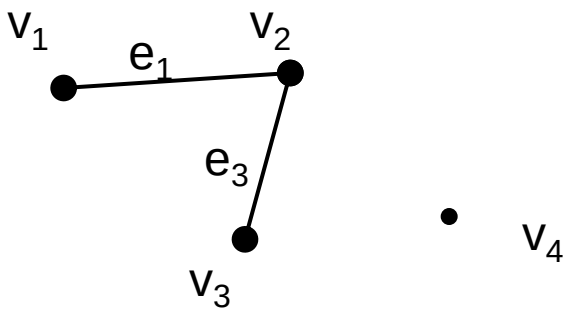


Etkilenmiş alt graf örneği(3):



G'

Etkilenmiş alt graf değil. Olması için E9 ayrıtı alınmalı idi.



G_1'

G_1' Alt graf olup, etkilenmiş alt graf değildir.

İki Parçalı Graf: G=(V,E) bir graf olsun ve V kümesi $V=V_1 \cup V_2$ şeklinde iki kümeye ayrıldığında,

- V_1 kümesindeki hiçbir tepe çifti bir ayrıtla birleştirilmemiş ise
- V_2 kümesindeki hiçbir tepe çifti bir ayrıtla birleştirilmemiş ise

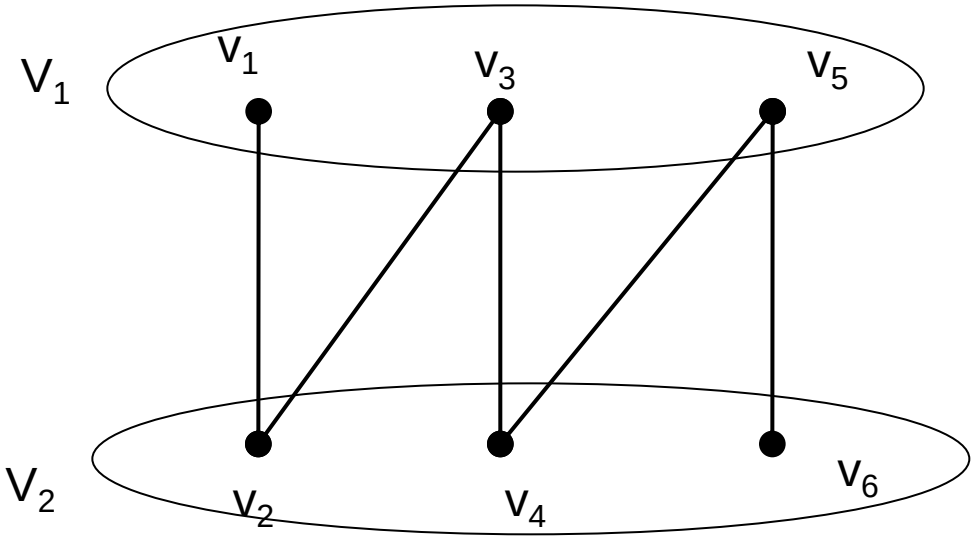
G ye iki parçalı graf denir.

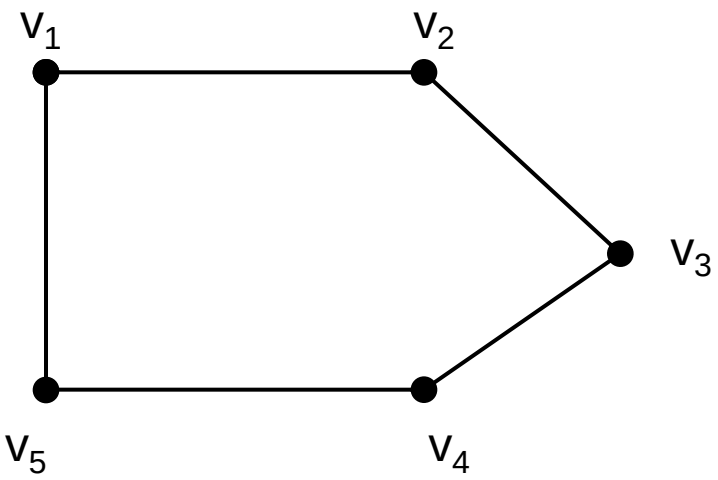
$V=\{v_1,v_2,v_3,v_4,v_5,v_6\}$

$V_1=\{v_1,v_3,v_5\}$

$V_2=\{v_2,v_4,v_6\}$

$\Rightarrow P_6$ iki parçalı bir graftır.





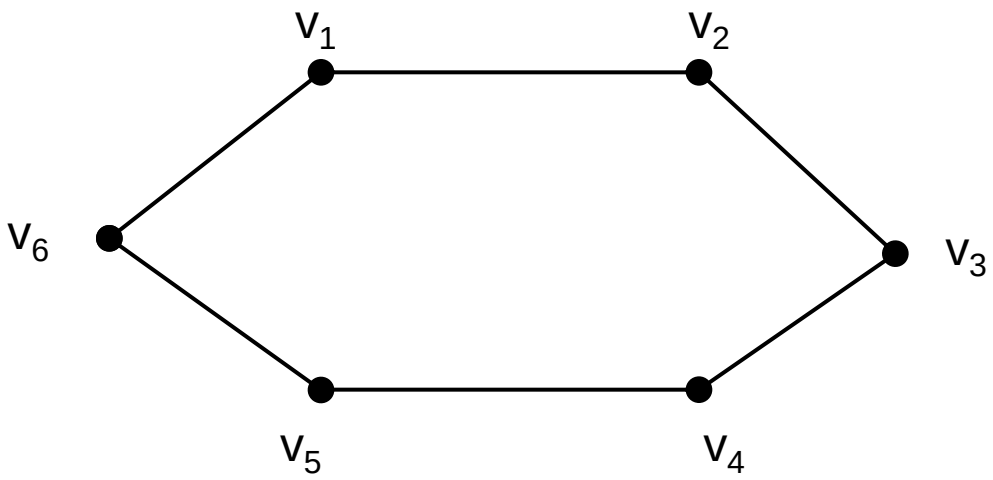
C_5

$V=\{v_1,v_2,v_3,v_4,v_5\}$

$V_1=\{v_1,v_3\}$

$V_2=\{v_2,v_4,v_5\}$

C_5; iki parçalı graf değildir.



C_6

$V=\{v_1,v_2,v_3,v_4,v_5,v_6\}$

$V_1=\{v_1,v_3,v_5\}$

$V_2=\{v_2,v_4,v_6\}$

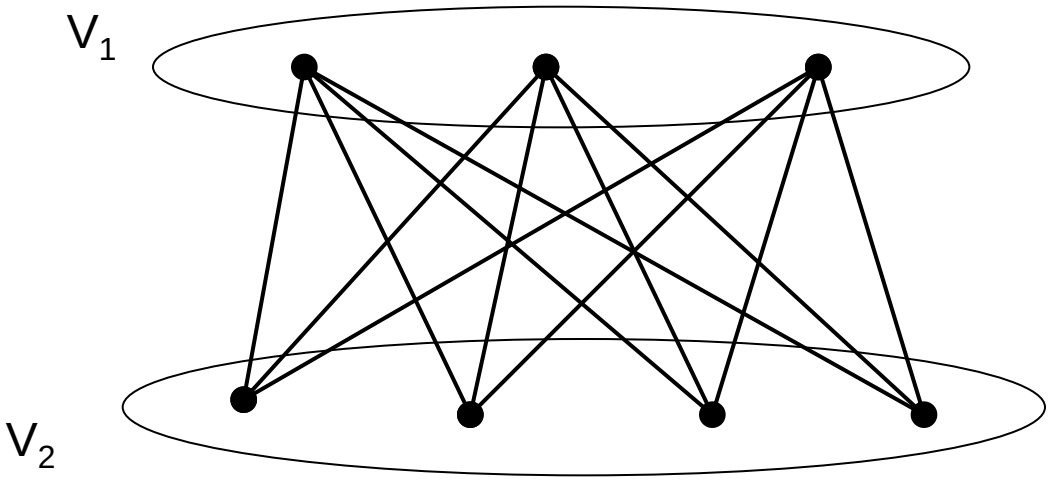
C_6; iki-parçalı graftır.

Sonuç: Her $P_n (n \geq 2)$ yol grafı 2-parçalı graftır.

Sonuç: Her C_n(n çift) çevre grafı 2-parçalı graftır.

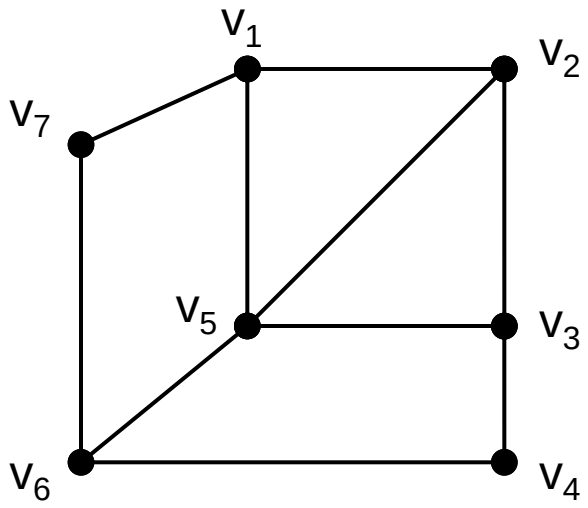
Sonuç: Her C_n(n tek) çevre grafı 2-parçalı graf değildir.

İki Parçalı Tam Graf: G iki parçalı bir graf, V tepeler kümesi $V=V_1 \cup V_2$ olsun V_1 deki her bir tepe V_2 deki her bir tepeye bir ayrıtla birleştirilmiş ise bu grafa iki-parçalı tam graf denir. $V_1=m$, $V_2=n$ olmak üzere iki-parçalı bir tam graf $K_{m,n}$ ile gösterilir.

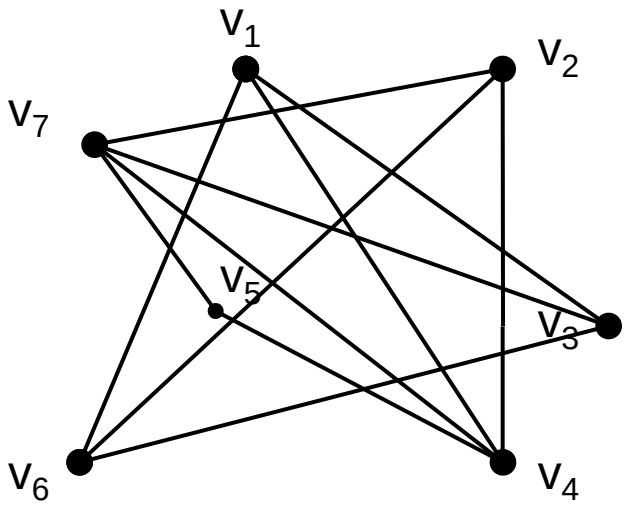


$K_{3,4}$

Tümleyen Graf: G(V,E) bir graf olsun. G grafının tümleyen grafı G'(V',E') ile gösterilir. Burada V'=V dir. E' kümesi ise G grafında olmayan ayrıtların oluşturduğu graftır.

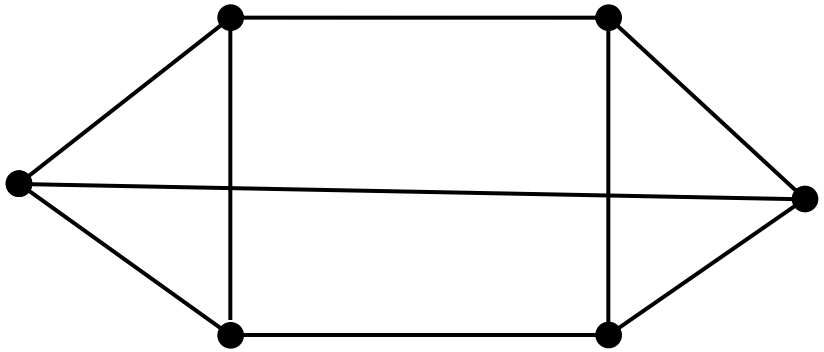


G



G'

r düzenli (r-regular) graf: Bir G grafının tüm tepelerine ait dereceler aynı ise bu grafa düzenli graf denir. Çevre graf ile tam graf regüler graflara örnek graflardır.

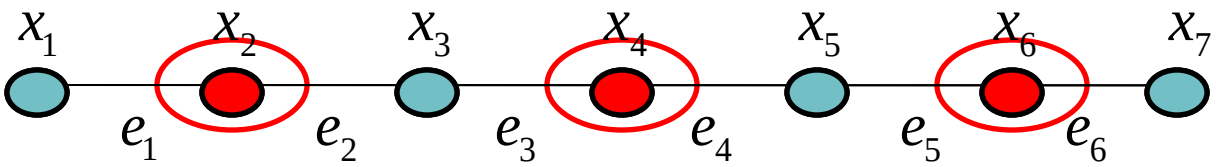


3-düzenli graf

GRAF PARAMETRELERİ

Bağımsızlık Sayısı: $S \subseteq V(G)$, bir grafının tepeler kümesinin herhangi bir alt kümesi olsun. S kümesindeki tepeleri ikişerli aldığımızda bu tepeler, G grafında bir ayrıta sahip değilse S kümesine **bağımsız küme** denir. Bir G grafı birden fazla bağımsız kümeye sahip olabilir. Bu kümeler içinde en çok elemana sahip olan kümenin eleman sayısına G grafının **bağımsızlık sayısı** (independence number) denir ve $\beta(G)$ ile gösterilir.

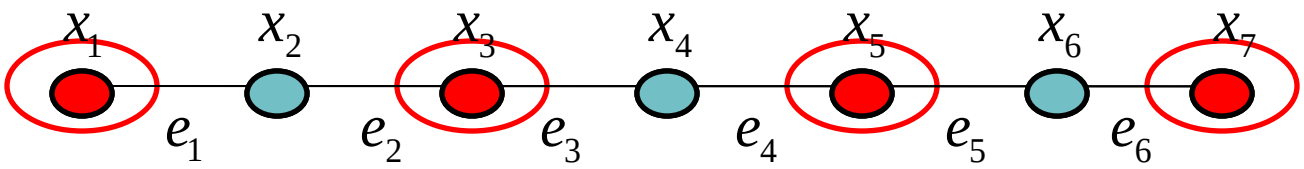
Örnek 1: P_7 grafının bağımsızlık sayısını bulalım.



$$S_1 = \{x_2, x_6\}$$

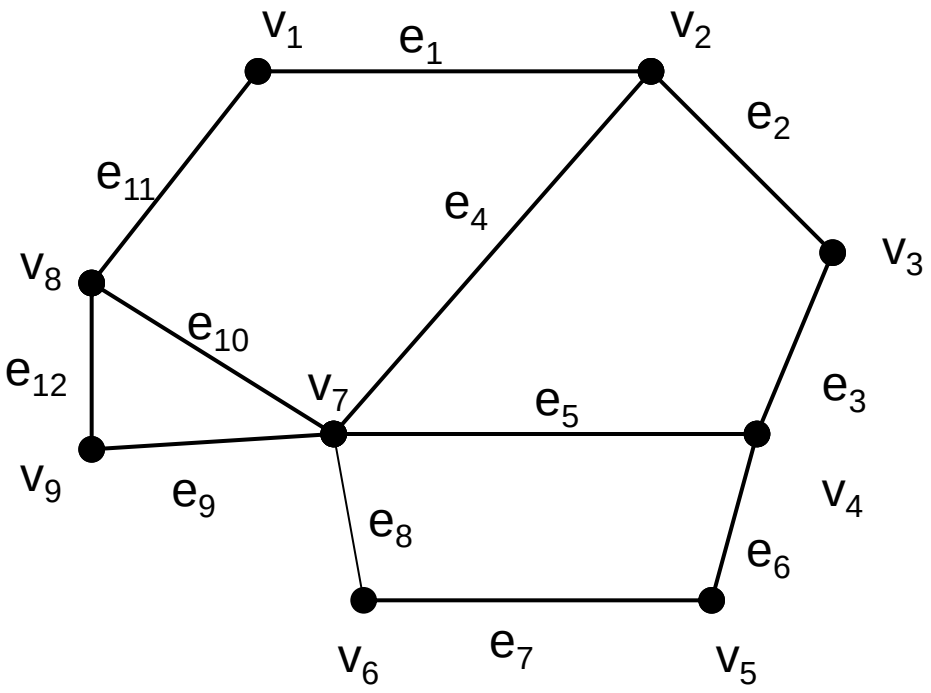
$$S_2 = \{x_2, x_6, x_4\}$$

3’ den daha büyük elemanlı bağımsız küme bulunabilir mi?



$$S_3 = \{x_1, x_3, x_5, x_7\}$$

S_1, S_2 ve S_3 bağımsız kümelerdir. $\beta(P_7) = 4$ dir.



G

$S=\{v_1,v_2\}$ $\Rightarrow$ Bağımsız küme değil.

$S=\{v_1,v_3\}$ $\Rightarrow$ Bağımsız küme.

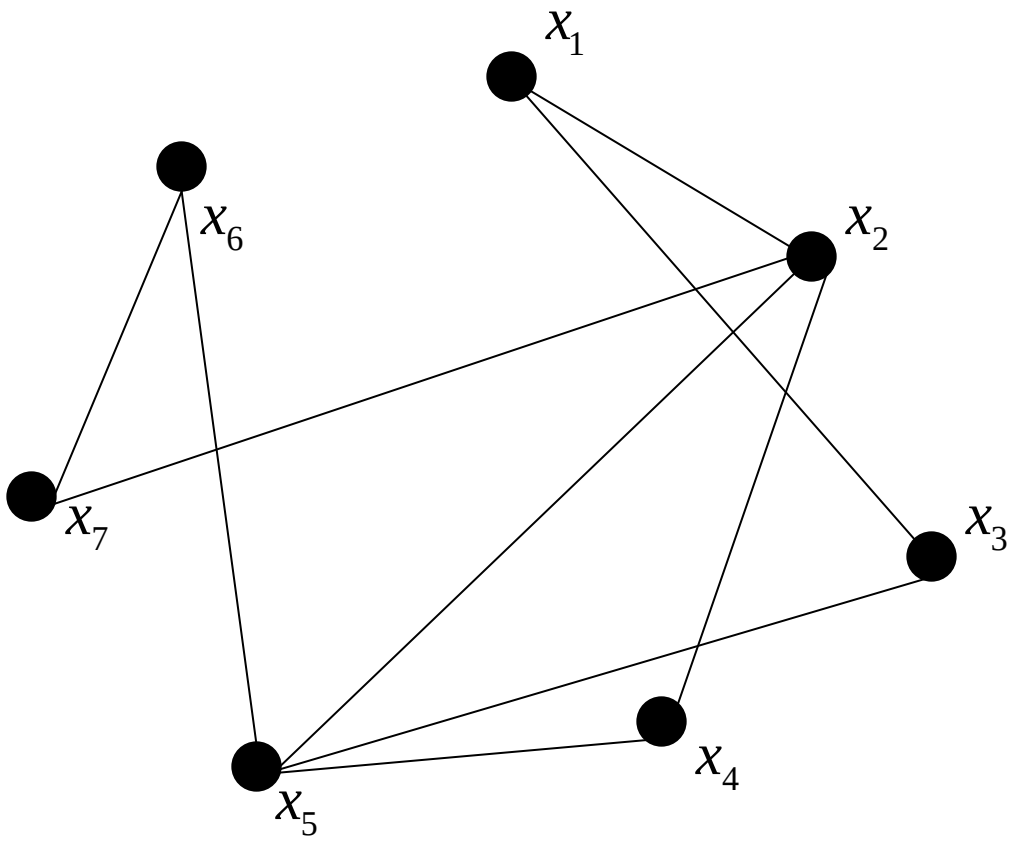
$S=\{v_1,v_3,v_5\}$ $\Rightarrow$ Bağımsız küme.

$S=\{v_1,v_3,v_5,v_7\}$ $\Rightarrow$ Bağımsız küme.

$S=\{v_2,v_4,v_6,v_8\}$ $\Rightarrow$ Bağımsız küme.

$\beta(G)= 4$ tir.

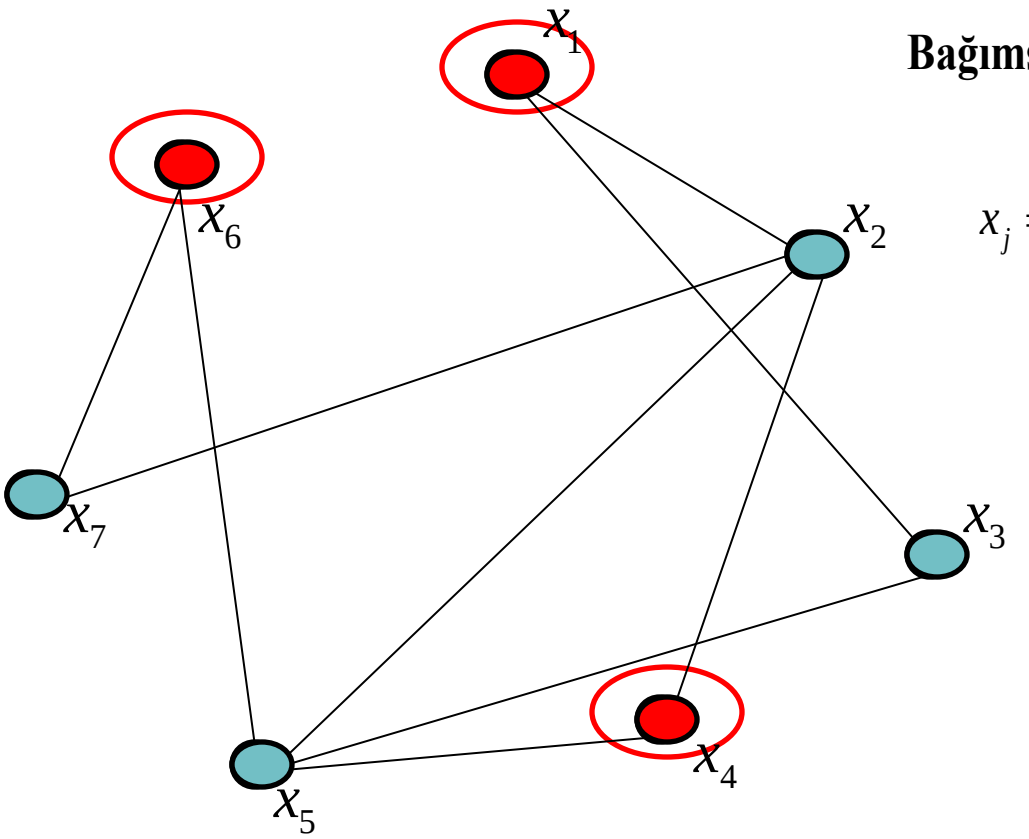
Problem : Bir askeri malzeme deposunda bazı malzemelerin yan yana bulunması durumunda kimyasal tepki verdikleri bilinmektedir. **Şekil 1**' deki grafta malzemeler birer tepe, kimyasal tepkiye giren malzemeler ise birer ayrıtla ifade edilmiştir. Bu malzemeleri hiçbir problem olmaksızın bir depoya yerleştirebileceğimiz bir plan hazırlayabilir miyiz?



Şekil 1

Burada açıktır ki, x_1 ve x_4 yanyana gelebilir. Fakat x_1 ve x_2 yanyana gelemez.

Bağımsız Küme Bulmanın Matematiksel Modeli



$$x_j = \begin{cases} 1 & , x_j \text{ tepesi seçilmiş ise} \\ 0 & , x_j \text{ tepesi seçilmemiş ise} \end{cases}$$

$$x_1 + x_2 \le 1$$
$$x_1 + x_3 \le 1$$
$$x_2 + x_4 \le 1$$
$$x_2 + x_5 \le 1$$
$$x_2 + x_7 \le 1$$
$$x_3 + x_5 \le 1$$
$$x_4 + x_5 \le 1$$
$$x_5 + x_6 \le 1$$
$$x_6 + x_7 \le 1$$

$$x_1 = 1 \quad x_2 = 0$$
$$x_4 = 1 \quad x_3 = 0$$
$$x_6 = 1 \quad x_5 = 0$$
$$x_7 = 0$$

$$k.a., \; Z_{\max} = \sum_{j=1}^{n} x_j$$

$$\sum_{j=1}^{7} x_j = 3 \Rightarrow \beta(G) = 3$$

Bağımsız küme problemi,

• Yerleştirme(Depolama) problemlerinin çözümünde

• Bağımsız kümeyi bulan bir çok algoritma vardır. En önemli algoritma Paull-Unger Algoritmasıdır.

Paull - Unger Algoritması

Paul-unger algoritması bir graftaki tüm maksimal bağımsız kümeleri ve bağımsızlık sayısını bulur.

Tanım 2.2.1 (Prather, 1976) $\sum=\{\sigma_1,\sigma_2,\ldots\sigma_n\}$ alfabesi ile $x=\sigma_{i_1}\sigma_{i_2}\ldots\sigma_{i_k}$ gibi kelimeleri oluşturabiliriz. (ε, null uzunluğu sıfır olan bir kelimedir). $\sum^*$ gösterilimi verilen $\sum$ alfabesinden üretilen bütün kelimelerin kümesinin bir yığınıdır.

$$\sum^*=\{x: x, \sum \text{ alfabesinin bir kelimesidir } \}$$

Şimdi bu yığını ilişkilendiren bir ikili işlem tanımlayacağız. Bu işlem genellikle kelimelerin **birleşimi**(bitişikliği) olarak adlandırılır. x yukarıda verildiği gibi ve $y=\sigma_{j_1}\sigma_{j_2}\ldots\sigma_{j_l}$ olsun. Birinci kelimeyi yazdıktan sonra ikinci kelimenin sembollerini yan yana dizeriz

$$x.y = \sigma_{i_1}\sigma_{i_2}...\sigma_{i_k}\ \sigma_{j_1}\sigma_{j_2}...\sigma_{j_l}$$

görüldüğü gibi her x kelimesi için

$$x.\varepsilon=\varepsilon.x=x$$

öyleyse $\sum^* = (\sum^*,\cdot,\varepsilon)$ bir monoid'e dönüştüğü görülür çünkü bitişiklik işleminin birleşmeli olduğu açıktır. $\sum^*$'yi $\sum$ alfabesinden elde edilen **free monoid** olarak adlandırırız.

Örnek 2.2.1: Eğer $\sum = \{0,1\}$ ise

$\sum^* = \{0,\ 1,\ 00,\ 01,\ 10,\ 11,\ 000,\ 001,\ \ldots\}$ şeklindedir. $\sum^*$'deki bitişiklik işleminin bazıları aşağıdaki gibi gösterilebilir.

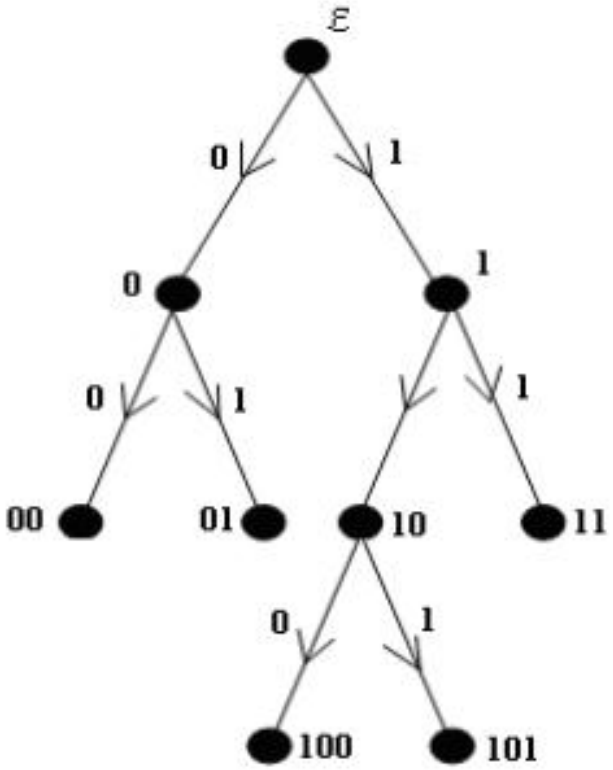
$$10\cdot001 = 10001$$

$$111\cdot01 = 11101$$

$$10\cdot\varepsilon = 10$$

ALGORİTMA

$x\ E\ y \Leftrightarrow y = x\sigma\ (\sigma \in \Sigma)$ ayrıt bağıntısı aracılığı ile birleştirilmiş bir sonsuz (digraf) $G = (\Sigma^*, E)$ grafının tepeler kümesi olarak, G grafının tepelerinin alfabesinden üretilen bütün kelimelerin kümesini Σ^* ile gösterelim. **(Prather, 1976).**



Şekil 1.a

Şekil **1.a' da** B={0,1} ikili alfabesinden oluşturulan B^* için sonsuz bir yönlendirilmiş ağaç graf gösterilmiştir. Şekil **1.a' da** görüldüğü gibi yönlendirilmiş dallar olarak gösterilen ayrıtları soldan sağa doğru etiketleriz, grafın bütününde B^*'ı temsil eden bu sıralamayı gerçekten incelemek istiyorsak bu gerekli olacaktır. Bu yönlendirilmiş ağaç grafta ilk tepe grafın kök tepesi olarak adlandırılır. Bu aynı zamanda Σ^* kümesindeki null kelimesidir.

V, B^*'ın sonlu bir alt kümesi olmak üzere T = (V, E) şeklinde sonsuz (digraf) bir ikili ağaçtır,

(i) $x\sigma \in V \Rightarrow x \in V\ (\sigma \in B)$

(ii) $x\ E\ y \Leftrightarrow$ bazı $\sigma \in B$ için $y = x\sigma$

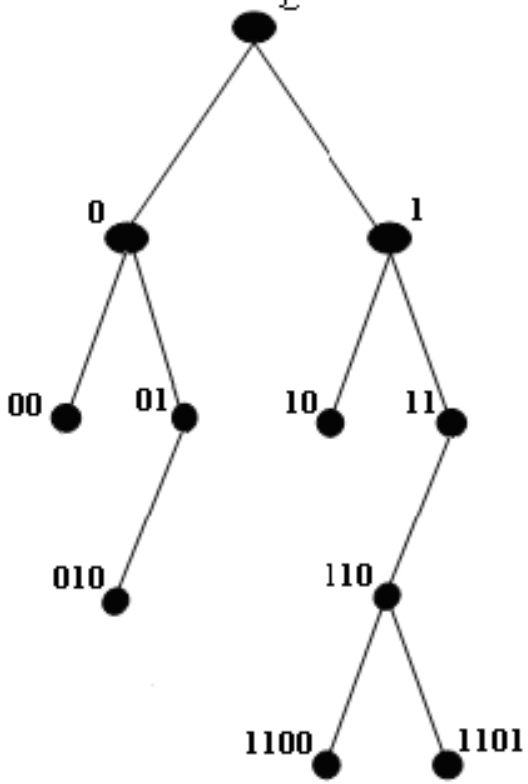
(i). madde bize her bir tepeden köke birleşmiş bir yol olduğunu garanti eder. *(ii)*. madde ise daha önce söylendiği gibi ayrıt bağıntısıdır. Aynı zamanda ağacın uçlarındaki düğümlere (tepelere) ait olan kümeleri gösteren alt küme aşağıdaki şekilde tanımlanır.

$$L = L(T) = \{x \in V: \text{bütün } \sigma \in B \text{ için } x\sigma \notin V \} \subseteq V$$

Örnek : Şekil 1.a' da bir ikili ağaç gösterilmektedir.

İkili ağacın tanımındaki (*i*). madde aşağıdaki şekilde örneklendirilmiştir.

$$1101 \in V \Rightarrow 110 \in V \Rightarrow 11 \in V \Rightarrow 1 \in V \Rightarrow \varepsilon \in V$$



Şekil 1.b.

Graftaki her bir tepenin gösterdiği x kelimesinin ($x \in \Sigma^*$) önce gelenlerinin grafta daha önceden var olduğu (*i*). madde tarafından garanti edilir. Örneğin, **Şekil 1.b.** ' deki 1101 graf tepesi 1101=110·1 şeklinde oluşturulmuş olup bu tepenin önce geleni 110' dır ve bu grafta daha önceden grafın tepesi olarak vardır. Bu özel T ağacı aşağıdaki düğümler (tepeler) kümesine sahiptir.

$$L = L(T) = \{ 00, 10, 010, 1100, 1101 \}$$

T = (V,E) ikili ağacı algoritmik işlemler içerisinde kullanılabilir. Sadece $V \subseteq B^*$ 'daki sonlu sayıdaki tepeler kullanılmalıdır. Bu yüzden algoritmada T için en uygun veri tipi dizi veri yapısıdır

T: array B^* of A

Burada A, tepeler etiketlenirken kullanılan cebirdir. B^* sonsuz bir küme olduğu için algoritmanın çalıştırılması sırasında bu tepelerin hepsini etiketlendirmek imkânsızdır. Sadece $V \subseteq B^*$ bir sonlu alt kümesi etiketlendirilecektir. Tepelerin etiketlendirildiği yukarıdan aşağıya doğru sıradan dolayı V alt kümesinin (*i*) durumu sağladığını söyleriz. Yani işlem sonlandığı zaman etiketlendirilmiş bir ikili ağaç elde ederiz.

Etiketlendirilmiş ikili ağaçlar, bir G grafının tüm maximal bağımsız kümelerinin bulunmasında ve aynı zamanda β bağımsızlık sayısının hesaplanmasında kullanılır. Bu hesaplamalar ilk M.C.Paull ve S.H.Unger tarafından yapılmıştır. Onların oluşturduğu çözümü aşağıdaki algoritma ile gösterebiliriz.

```
T: array B* of 𝒫(V)          β, i, j, n: positive integer
E: array n+ × n+ of B              L, M: subset of B*
v: array B of V                    x: element of B*
                  σ: element of B*
begin  T ← ∅; T[ε] ← V; L ← {ε};
       for j ←1 to n – 1 do
       for i ←j +1 to n do
       if E[i, j] = 1 then
       begin M ← {x ∈ L: {v_i, v_j}⊆ T[x]}; L ← L ~ M;
```

$v[0] \leftarrow v_i$; $v[1] \leftarrow v_j$;

for $x \in M$ **do**

for $\sigma \in B$ **do**

begin $T[x\sigma] \leftarrow T[x] - \{v[\sigma]\}$;

if $((T[x\sigma] \not\subset T[y])$ **for all** $y \in L$ **then**

$$L \leftarrow L \cup \{x\sigma\}$$

end;

end;

$$\beta \leftarrow \max_{x \in L} \{|T[x]|\}$$

End.

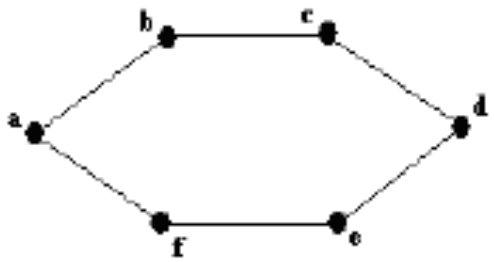
Bu algoritma herhangi bir *G* grafı için β bağımsızlık sayısını belirler ve aynı zamanda bütün maksimal bağımsız kümelerin listesini verir. Bunlardan daha büyük bir bağımsız küme içerilmez.

Örnek : Paull-Unger algoritmasındaki çeşitli adımları sırasıyla örneklemek için **Şekil 1.c.** deki C_6 grafını inceleyelim. İlk olarak **Şekil 1.d.** (a)' da gösterildiği gibi yalnızca ağacımızın kökünü etiketleyelim.

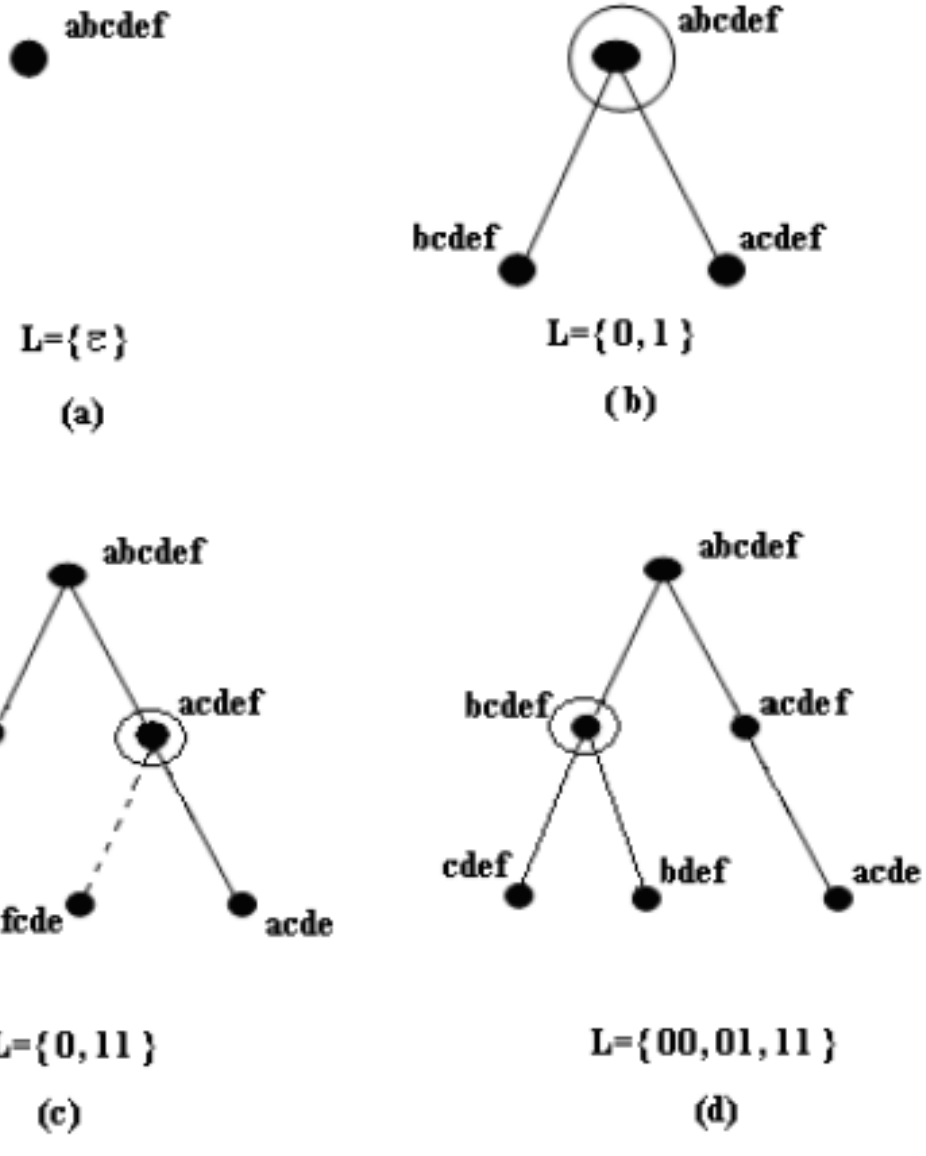
Bu noktada L ={ε} dir. Daha sonra alt üçgensel matristeki girdileri inceleyebiliriz.

Alt üçgensel matris,

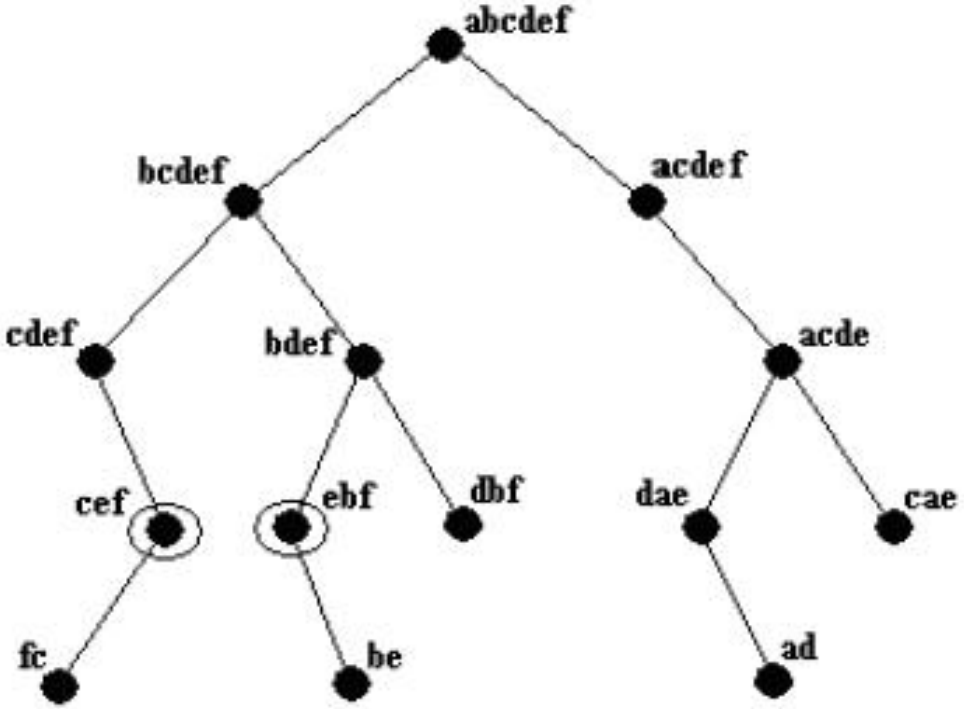
b	1				
c	0	1			
d	0	0	1		
e	0	0	0	1	
f	1	0	0	0	1
	a	b	c	d	e



Şekil 1.c.


abcdef
L={ε}
(a)
abcdef
bcdef
acdef
L={0,1}
(b)
abcdef
bcdef
acdef
fcde
acde
L={0,11}
(c)
abcdef
bcdef
acdef
cdef
bdef
acde
L={00,01,11}
(d)

Şekil 1.d.



Şekil 1.e.

Yuvarlak içine alınmış düğümler, matriste son olarak bulunmuş 1-girişi (eEf) tanımlandığı zaman işaretlenmiş M' lerdir. Bununla birlikte sonuç olarak,

L={0010, 0101, 011, 1101, 111} dir.

Böylece, $\beta = \max_{x \in L} \{|T[x]|\} = 3$ tür.

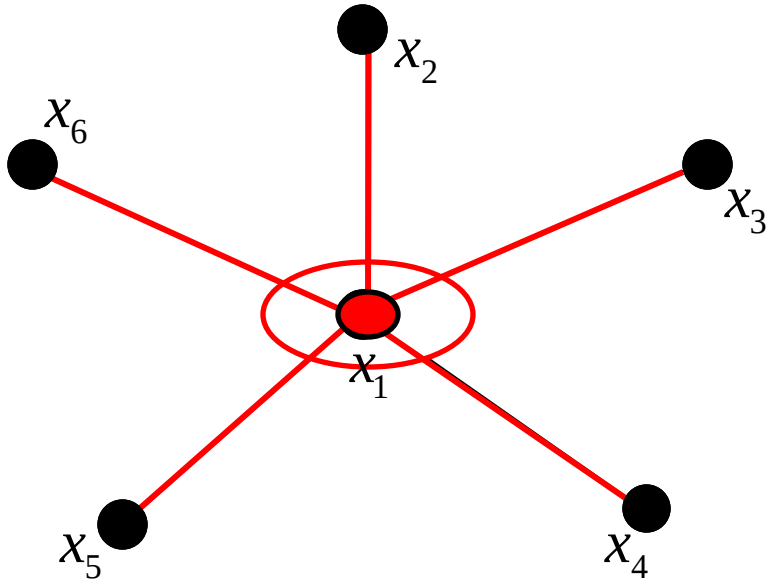
Algoritmanın uygulaması sonucunda elde edilen bütün dalların etiketleri maksimal bağımsız kümelerle eşleşir. Sırasıyla bütün bağımsız alt kümeler,

$$\{f, c\}, \{b, e\}, \{d, b, f\}, \{a, d\}, \{c, a, e\}$$

şeklindedir.

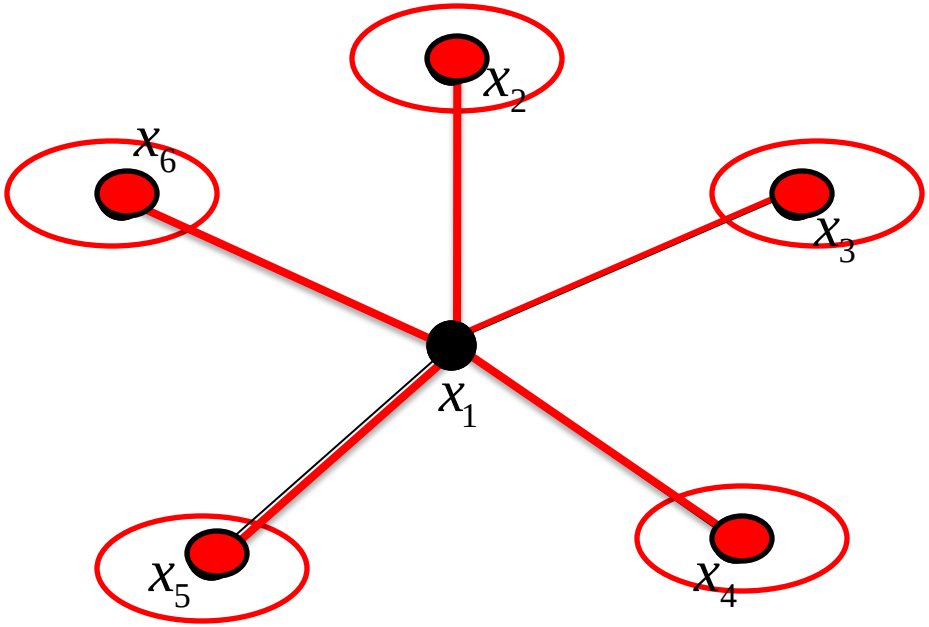
Örtü Sayısı: $S \subseteq V(G)$, bir G grafının tepeler kümesinin boş olmayan bir alt kümesi olsun. G grafındaki her bir ayrıtın en az bir uç noktası bu S kümesinde ise bu kümeye **örtü kümesi** denir. Bir G grafı birden çok örtü kümesine sahip ola bilir. Bu kümeler içinde en az elemana sahip olan kümenin eleman sayısına G grafının **örtü sayısı** (covering number)denir ve $\alpha(G)$ ile gösterilir.

Örnek 4.2: $K_{1,5}$ grafının örtü sayısını bulalım.



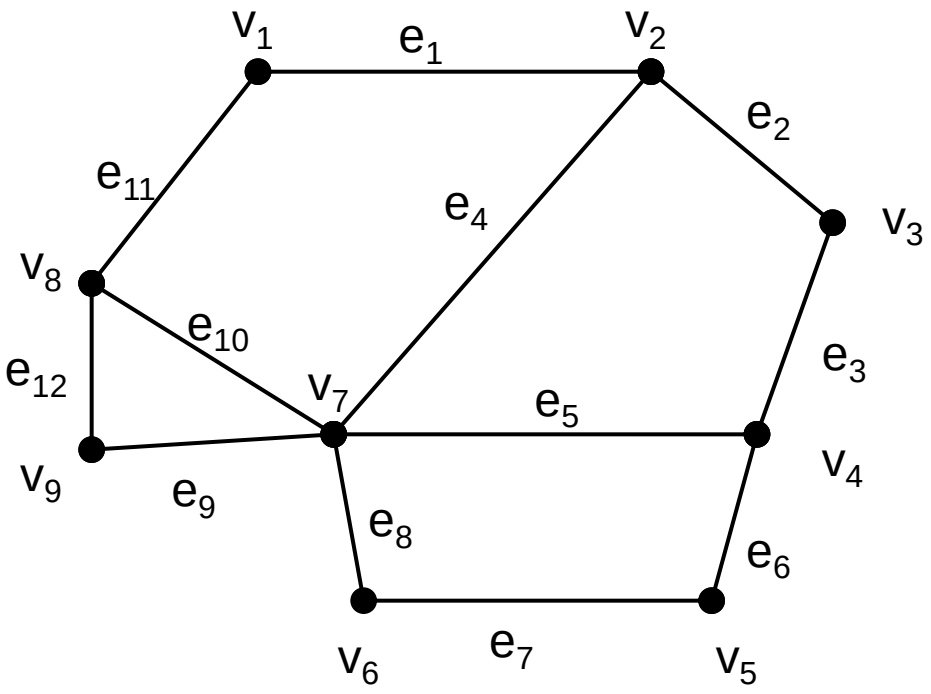
$$S_1 = \{x_1\}$$

Benzer şekilde başka bir örtü kümesi,



$$S_2 = \{x_2, x_3, x_4, x_5, x_6\}$$

S_1 ve S_2 örtü kümeleridir. $K_{1,5}$ grafının örtü sayısı $\alpha(K_{1,5}) = 1$ dir



$V(G)=\{v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8,v_9\}$

$S=\{v_1,v_3,v_5\}$ $\Rightarrow$ Örtü kümesi değildir.

$S=\{v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8\}$ $\Rightarrow$ Örtü kümesidir.

$S=\{v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8,v_9\}$ $\Rightarrow$ Örtü kümesidir.

$S=\{v_2,v_4,v_6,v_7,v_8\}$ $\Rightarrow$ Örtü kümesidir.

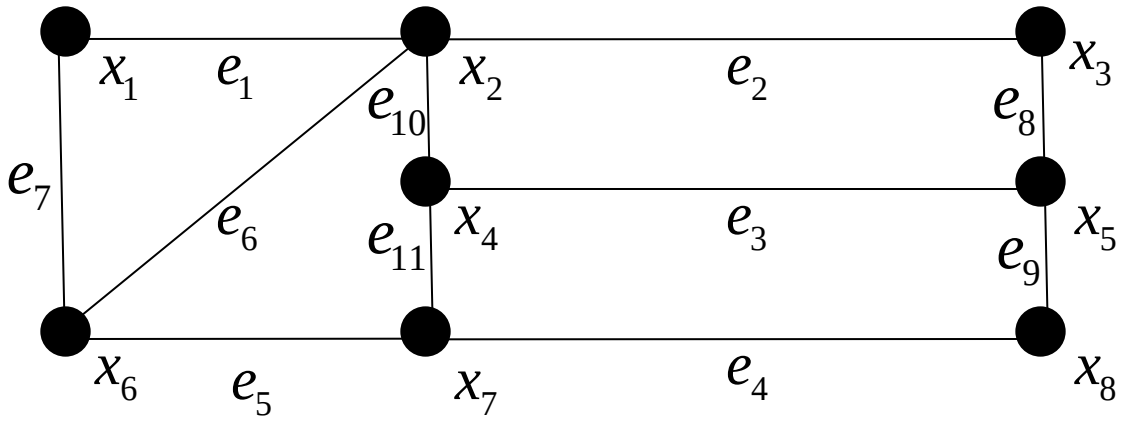
En az elemanlı kümenin eleman sayısı=5 dir ve $\alpha(G)= 5$ tir.

Örtü kümesi problemi,

- Güvenlik problemlerinin çözümünde
- Yerleştirme Problemlerinin(Ambulans, İtfaiye) çözümünde

Problem 4.2: Öğrencilerin gün içi ihtiyaçlarını sağlamak için bir üniversite kampüsüne acil telefonlar yerleştirilmek isteniyor. **[** Şekil 4.2' kampüs bir graf ile modellenmiştir.**]**

Bununla beraber kampüsün her ana sokağına en azından bir telefonla hizmet getirilmek isteniyor. Bu iş için **<u>en az</u>** kaç telefona ihtiyacımız vardır?



G grafı

Şekil 4.2

Problemin Matematiksel Modeli

$$x_j = \begin{cases} 1, & j.ci\ yere\ telefon konursa \\ 0, & j.ci\ yere\ telefon\ konmazsa \end{cases}$$

e_1 sokağı için, $x_1 + x_2 \geq 1$

e_2 sokağı için, $x_2 + x_3 \geq 1$

e_3 sokağı için, $x_4 + x_5 \geq 1$

e_4 sokağı için, $x_7 + x_8 \geq 1$

e_5 sokağı için, $x_6 + x_7 \geq 1$

e_6 sokağı için, $x_2 + x_6 \geq 1$

e_7 sokağı için, $x_1 + x_6 \geq 1$

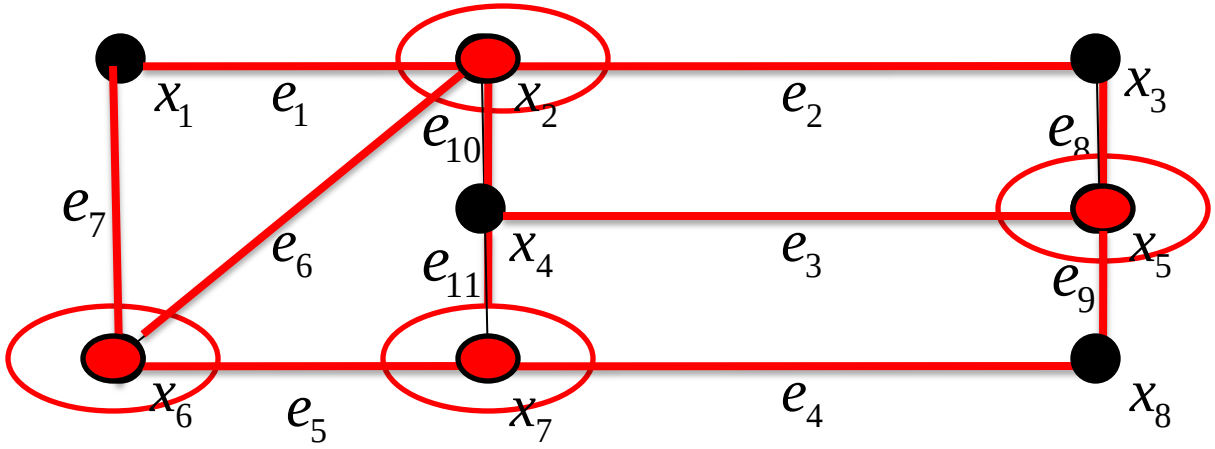
e_8 sokağı için, $x_3 + x_5 \geq 1$

e_9 sokağı için, $x_5 + x_8 \geq 1$

e_{10} sokağı için, $x_2 + x_4 \geq 1$

e_{11} sokağı için, $x_4 + x_7 \geq 1$

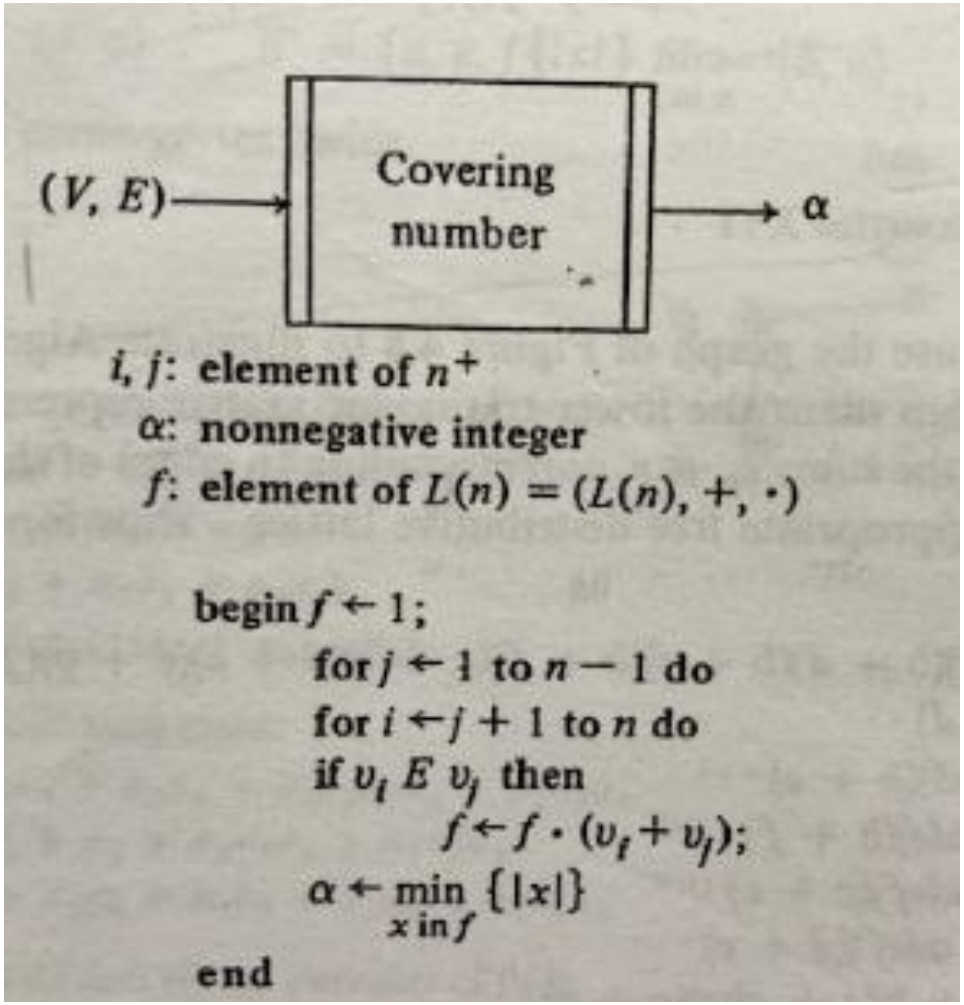
k.a., $Z_{min} = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8$



G grafı

$$S = \{x_2, x_5, x_6, x_7\} \Rightarrow \alpha(G) = 4$$

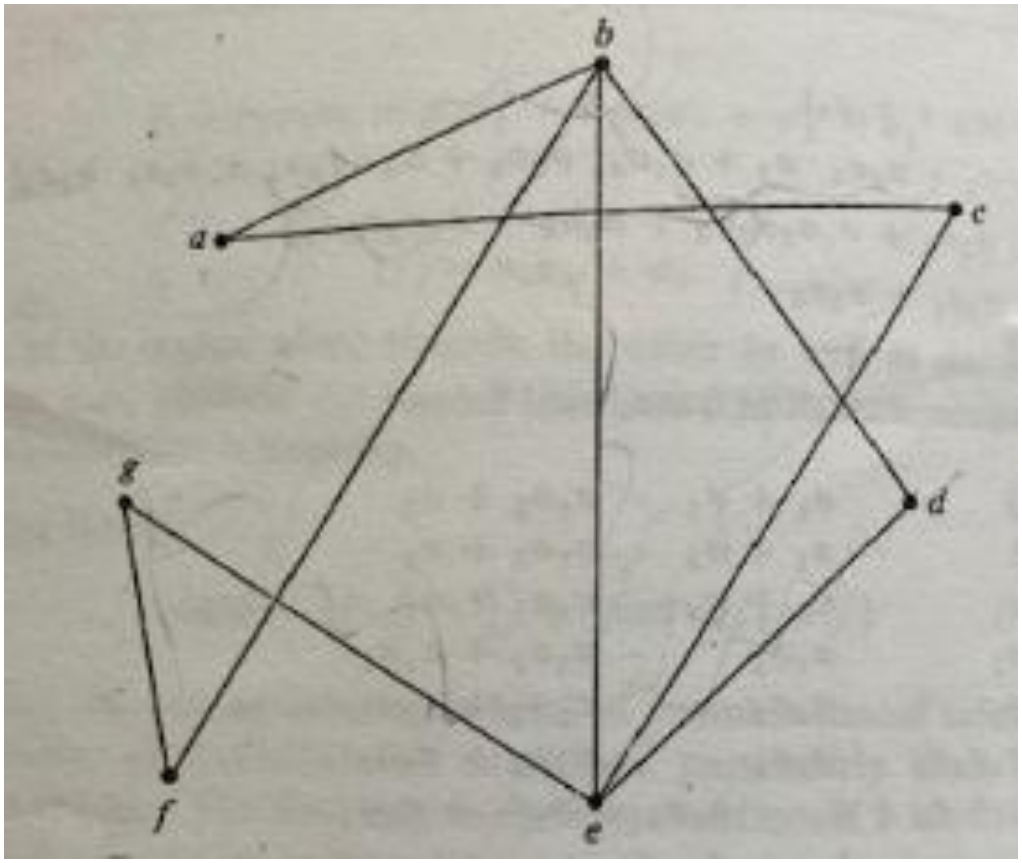
Örtü Sayısı bulmak için bir algoritma



i, j: element of n^+
α: nonnegative integer
f: element of $L(n) = (L(n), +, \cdot)$

```
begin f ← 1;
      for j ← 1 to n − 1 do
      for i ← j + 1 to n do
      if v_i E v_j then
            f ← f · (v_i + v_j);
      α ← min_{x in f} {|x|}
end
```

Örnek:



Şekilde verilen grafın örtü kümelerini ve örtü sayısını bulunuz.

$$
\begin{aligned}
f &= (a + b)(a + c)(b + d)(b + e)(b + f)(c + e)(d + e)(e + g)(f + g) \\
&= (a + bc)(b + d)\cdots \\
&= (ab + bc + ad)(b + e)\cdots \\
&= (ab + bc + ade)(b + f)\cdots \\
&= (ab + bc + adef)(c + e)\cdots \\
&= (bc + abe + adef)(d + e)\cdots \\
&= (bcd + adef + bce + abe)(e + g)\cdots \\
&= (adef + bce + abe + bcdg)(f + g) \\
&= adef + bcef + abef + bceg + abeg + bcdg
\end{aligned}
$$

Böylece örtü sayısı 4 bulunur.

Teorem: n- tepeli bir G grafı için $\alpha(G)+\beta(G)=n$ dir.

İspat: ???

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