8 Graf Teorisi

8.1 Graflar ve Tanımlar

Graf teorisinin uygulamaları modern hayatın karmaşık ve geniş kapsamlı birçok probleminin çözümü için kullanılmaktadır. Bu uygulamalar; ekonomi, yönetim bilimi, satış pazarlama, bilgi iletimi, taşıma planlaması gibi alanları kapsamaktadır. Graf teorisi problemleri tanımlama ve yapısal olarak ilişkileri belirlemekte de faydalıdır.

Bir grafın ne olduğunu açıklamadan önce belki de ne olmadığını söylemek daha iyi olabilir. Bu bölümde kullanılan graf bir fonksiyonun grafiği değildir. O halde graf nedir? Basitçe bir graf düğüm olarak adlandırılan noktalar ve her biri bu noktaları veya sadece noktanın kendisini birleştiren ve ayrıt olarak adlandırılan çizgiler topluluğudur. Örnek olarak şehirleri düğüm(vertice) ve onları bağlayan yolları ayrıt(edge) olarak gösteren yol haritaları verilebilir.

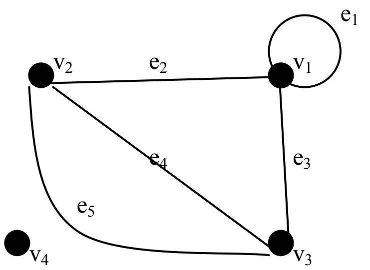
Bir grafı tanımlamak için öncelikle düğümlerin ve ayrıtların kümesini tanımlamak gerekir. Daha sonra hangi ayrıtların hangi düğümleri bağlandığı belirtilmelidir. Bir ayrıt her iki ucunda da bir düğüm olacak şekilde tanımlandığından graftaki tüm ayrıtların uç noktalarını bir düğüm ile ilişkilendirmek gerekir. Bu nedenle, her bir e ayrıt'ı için $\{v_1, v_2\}$ kümesi tanımlarız. Bunun anlamı e ayrıt'ının v_1 ve v_2 düğümlerini **bağladığı**dır. $v_1 = v_2$ olabilir. $\{v_1, v_2\}$kümesi δ(e) ile gösterilir ve düğümler kümesinin bir alt kümesidir.

Tanım: Bir **yönsüz (undirected) graf G** şunlardan oluşur: **G**(V,E)

(i) boş olmayan sonlu bir V düğümler kümesi

(ii) sonlu bir E ayrıtlar kümesi ve

(iii) bir $\delta:E\rightarrow P(V)$ fonksiyonu öyle ki her bir e ayrıtı için δ(e) V 'nin bir veya iki elemanlı bir alt kümesidir.



Şekil 7.1

Şekil 7.1 ' deki G grafına bakalım. Açıktır ki, G grafının düğüm kümesi $\{v_1, v_2, v_3, v_4\}$ ve ayrıt kümesi $\{e_1, e_2, e_3, e_4, e_5\}$. $\delta:E\rightarrow P(V)$ fonksiyonu şöyle tanımlanmaktadır:

$$\delta: e_1\{v_1\}$$

$$\delta: e_2\{v_1,v_2\}$$

$$\delta: e_3\{v_1,v_3\}$$

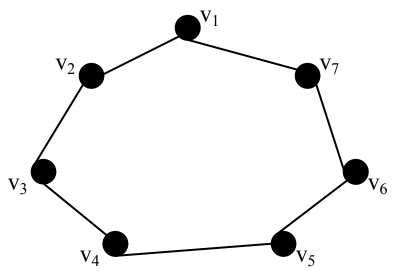
$$\delta: e_4\{v_2,v_3\}$$

$$\delta: e_5\{v_2,v_3\}$$

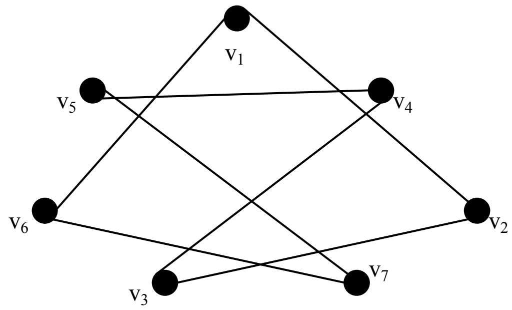
Bu basitçe, e_1' in v_1 düğümünü kendisine, e_2'nin v_1 ve v_2 düğümlerini vs. bağladığını gösterir.

Yukarıda görüldüğü gibi bir ayrıt bir düğümü yine kendisine bağlayabileceği (loop) gibi, bir düğüm hiçbir ayrıt ile bağlanmamış olabilir (v_4' te olduğu gibi). Ayrıca iki düğüm birden fazla ayrıt ile de (multiple ayrıt) bağlanmış olabilir.

Dikkat edilmesi gereken bir nokta bir graf ile onu temsil eden diyagram aynı değildir. Daha önce de söylediğimiz gibi bir graf bir fonksiyon ile birlikte iki kümeden oluşur. Şekil 7.1 kendi başına bir graf değildir sadece bir grafın gösterimidir. Verilen bir graf birbirinden çok farklı görünen iki graf ile gösterilebilir. Örneğin şekil 7.2 'deki iki diyagram çok farklı görünmelerine rağmen aynı G grafını temsil ederler. Şekil 7.2(a) 'daki graf 7 düğümlü çember (Wheel) graf olarak adlandırılır ve W_7 şeklinde gösterilir. Tüm n pozitif tamsayıları için n düğümlü ve n ayrıtlı bir W_n çember grafı vardır.



Şekil 7.2 (a)

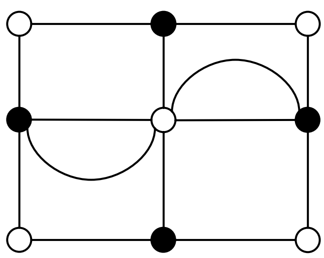


Şekil 7.2 (b)

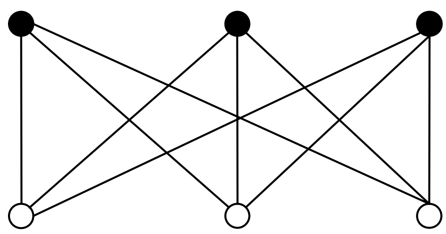
Eğer bir graf bir düğümü yine kendisine bağlayan (çevrim içermeyen) bir ayrıt ve aynı iki düğümü birden fazla bağlayan ayrıtlara sahip değilse **basit** (simple) graftır. Eğer grafta her bir ayrıta bir gerçel sayı atanmış ise graf bir ağ dır veya ağırlıklı graftır.

Bir yönlü graf (veya digraf) G=(V,E), burada V sonlu düğümü kümesi ve E 'de sonlu yaylar kümesidir. Öyle ki, E'deki her bir e yayı v ve w sıralı düğümlerini birleştirir. e=(v,w) demek, e, v'den w'ye bir yaydır veya e v'den çıkar w'ye girer demektir. **Karma**(mixed) G=(V,E) grafta ise, E'nin en az bir elemanı kenar ve e'nin en az bir elemanı yaydır.

İki parçalı grafta, düğümler kümesi, iki ayrı kümeye parçalanabilir, öyle ki V_1'deki bir düğüm ve V_2' deki bir düğüm arasındaki her bir ayrıt $G=(V_1,V_2;E)$ ile gösterilir. Eğer n düğümlü bir basit graf'ta her bir düğüm çifti arasında bir ayrıt var ise buna tam graf denir ve K_n ile gösterilir. Bir komple iki parçalı graf tamamen $|V_1|$ ve $|V_2|$ ile belirtilir. n ve m düğümlü komple iki parçalı graf $K_{n,m}$ ile gösterilir ve $|V_1|=n$ ve $|V_2|=m$ 'dir. Şekil 7.3 iki tane iki parçalı graf örneğidir. Her iki şekilde de V_1 'in düğümleri içi dolu noktalar ile, V_2 **'**nin düğümleri ise içi boş noktalar ile gösterilmiştir. (b)' deki graf komple iki parçalı $K_{3,3}$ grafıdır.



Şekil 7.3 (a)



Şekil 7.3 (b)

Tanım: **(i)** v ve w düğüm çiftini bağlayan bir e ayrıt' ı varsa bu iki düğüm komşudur (adjacent). Bu durumda hem v hem de w e' ye değer deriz ve ayrıca e de v ve w 'ya değer deriz.

(ii) $e_1, e_2, \ldots e_n$ ayrıtları en az bir ortak düğüme sahipse komşudur.

(iii) Bir v düğümünün derecesi (**degree**) $\sigma(v)$, v düğümüne bağlı olan ayrıtların sayısıdır. (Aksi belirtilmediği sürece v' yi kendisine bağlayan ayrıt v'nin derecesini iki arttırır.) Tüm düğümleri aynı r derecesine sahip grafa r dereceli düzenli (regular) graf veya r-derece denir.

(iv) Bir boş graf (**null**) veya tamamen bağlı olmayan (**totally disconnected**) graf ayrıt kümesi boş olan graftır.

(v) Bir tam (**complete**) graf ayrı düğüm çiftlerinin tümü bir ayrıt ile bağlı olan basit graftır ve n düğüm sayısı olmak üzere K_n şeklinde gösterilir.

(vi) Düğüm kümesi, tüm ayrıtların V_1 'in bir düğümünü V_2' nin bir düğümüne bağladığı $\{V_1, V_2\}$ şeklinde bir bölmelemeye sahip olan grafa iki parçalı (**biparite**) graf denir.

(vii) Bir komple iki parçalı (**complete bipartite**) graf V_1 'in tüm düğümlerini V_2' nin tüm düğümlerine tek bir ayrıt ile bağlayan iki parçalı graftır.

Örnek 7.1: G , V düğüm setinin $\{V_1, V_2\}$ bölmelemesine sahip olduğu bir iki parçalı graf olsun. Dikkat edilirse G basit graf olmak zorunda değildir. Gereken tek şey her bir ayrıt V_1 'in bir düğümü ile V_2' nin bir düğümünü bağlamalıdır. $v_1 \in V_1$ ve $v_2 \in V_2$ dersek, bunları bağlayan birden fazla ayrıt olabilir veya hiç ayrıt olmayabilir. Açıkça görülüyor ki, G'de çevrim yoktur.

G grafının farklı gözüken diyagramlar ile gösterilebileceğini belirtmiştik. Bir grafı göstermenin başka bir yolu da ileride tanımlayacağımız komşuluk matrisi (adjacency matrix) yardımıyla olur.

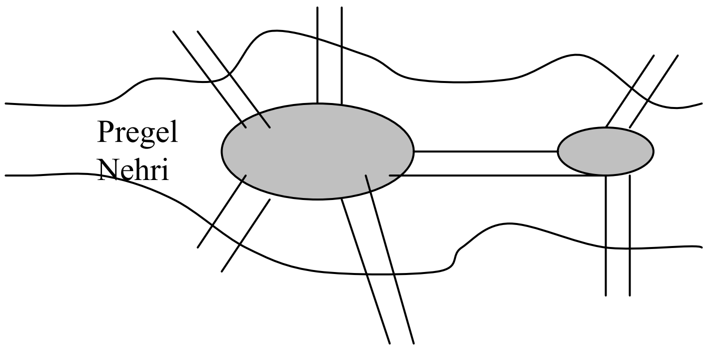
Tanım: Eğer, **V'**$\subseteq$V(düğüm kümesi), E'$\subseteq$E(Ayrıt kümesi) ve $\delta_{G'}(e)=\delta_G(e)$ ise, bir G' =(V',E') grafı G =(V,E) grafının alt grafıdır(subgraph) ve G' $\leq$ G şeklinde gösterilir.

G' grafının tüm e ayrıtları için $\delta_{G'}(e)=\delta_G(e)$ durumu G' alt grafının ayrıtlarının G de olduğu gibi aynı düğümleri bağlaması gerektiği anlamına gelir. Eğer G nin diyagramından bazı düğümleri veya ayrıtları silerek G' nin diyagramını elde edebiliyorsak G', G 'nin alt grafıdır. Tabii ki, bir düğümü siliyorsak ona çakışık olan tüm ayrıtları da silmeliyiz.

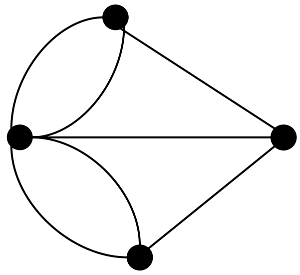
Örnek: Könnisberg Köprüsü problemi

Euler 1736 yılında yazdığı makale ile graf teorisinin doğmasına sebep olmuştur. Bu makale aşağıda tanımlanan Könnigsberg Bridge Problemlemini çözebilen bir teoriyi içeriyordu. Pregel nehri Königsberg kasabasının içinden akmaktadır. Nehrin ortasında şekil 7.4 (a) daki gibi nehrin kıyılarına ve birbirine köprüler ile bağlı iki ada bulunmaktadır. Königsberg kasabasının vatandaşları için problem kıyıların veya adaların birinden başlayıp tüm köprülerden sadece bir kez geçerek başladığımız yere yürüyebilir miyiz?

Euler öncelikle şekil 7.4 (b) ' deki gibi Königsberg coğrafyasının gerekli özelliklerini bir graf ile gösterdi. Her bir nehir kıyısı ve adalar bir düğüm ile köprüler de ayrıtlar ile temsil edildi. Graf teorisi terimleri ile problem şu hale geldi: grafın tüm ayrıtlarını içeren kapalı bir yol var mıdır?



Şekil 7.4 (a)



Şekil 7.4 (b)

Komşuluk ve Çakışım Matrisleri

Tanım: G düğüm kümesi $\{v_1, v_2,..,v_n\}$ olan bir graf olsun. G 'nın komşuluk matrisi; a_{ij}, v_i ve v_j 'yi bağlayan ayrı ayrıtların sayısı olmak üzere n x n A=A(G) matrisidir.

Komşuluk matrisi v_i ve v_j 'yi(düğüm) bağlayan ayrıtların sayısı, v_j ve v_i 'yi bağlayan ayrıtların sayısıyla aynı olduğundan simetrik olmalıdır. v_i düğümünün derecesi komşuluk matrisinden kolayca belirlenebilir. v_i de bir loop yoksa bu düğümün derecesi matrisin i. sütunundaki değerlerin toplamıdır. Her bir loop dereceyi iki kere etkilediğinden i. sütundaki değerleri toplarken a_{ii} diyagonal elemanın iki katı alınır.

Örnek 7.2: Aşağıdaki A komşuluk matrisi şekil 7.1 'de gösterilen grafa aittir.

$$A=\begin{vmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}.$$

$V=\{v_1, v_2, v_3, v_4\}$ ile A' nın satırları ve sütunları düğümleri temsil eder.

Grafın iki özelliği matrise bakılarak hemen görülebilir. Öncelikle, diyagonale bakıldığında bir tek çevrim(loop) vardır- v_1 ' den kendisine. İkincisi, son satır veya sütundaki 0' lar v_4 'ün bir tecrit edilmiş düğüm yani başka hiçbir düğüme bağlı olmayan (kendisi dahil) bir düğüm olduğunu gösterir.

Düğümlerin dereceleri matristen kolayca hesaplanabilir:

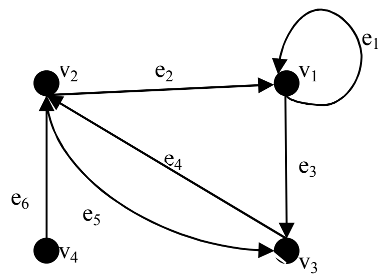
$\sigma(v_1) = 2.1+1+1=4$

$\sigma(v_2) = 1+2=3$

$\sigma(v_3) = 1+2=3$

$\sigma(v_4) = 0.$

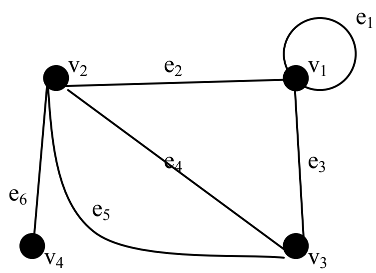
n ayrıtlı bir **yönlü grafın komşuluk matrisi** de nxn lik bir matristir $A=(a_{ij})$. Eğer i. düğümden den j. düğüme bir yay var ise $a_{ij}=1$ diğer durumda ise 0 dır.(Şekil 7.5)



$$\begin{matrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$

Şekil 7. 5: Yönlü Graf ve Komşuluk Matrisi

Graf ve yönlü graflarda diğer önemli bir matris ise çakışım matrisidir. Komşuluk matrisinin tersine çakışım matrisinde çoklu ayrıt ve paralel yaylar gösterilebilir. V={1,2,,…,n} ve E={e_1,e_2,….,e_m} olmak üzere G=(V,E) grafı verilsin. G grafının çakışım matrisi, nxm boyutlu olan ve her bir satırın bir düğüme ve her bir kolonun bir ayrıta karşılık geldiği bir B=(b_{ik}) matrisidir, öyle ki eğer e_k, i ve j. düğümler arasındaki bir ayrıt ise k. kolonun elemanlarından b_{ik}=b_{jk}=1, diğerleri sıfırdır. Çevrim olan ayrıtın kolonunda sadece bir tek 1 vardır. Eğer grafta çevrim yok ise düğümlerin derecelerinin toplamı ayrıt sayısının iki katına eşittir. Çünkü bu özellikteki ayrıtlar iki düğümü birbirine bağlarlar.(Şekil 7.6.)



$$\begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 1 & 1 & 0 & 0 & 0 \\ v_2 & 0 & 1 & 0 & 1 & 1 & 1 \\ v_3 & 0 & 0 & 1 & 1 & 1 & 0 \\ v_4 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

Şekil 7.6 Çakışım matrisi

Teorem 7.1: Eğer G, çevrim içermeyen ve m ayrıtlı bir çoklu graf ise, G'nin bütün düğümlerinin derecelerinin toplamı 2m'dir.

Bir yönlü grafın(çevrim içermeyen) çakışım matrisi, eğer e_k i den j'ye bir yay ise, k. kolondaki b_{ik}=-1 ve b_{jk}=1 diğer elemanlar sıfırdır.

Graflarda Bağlılık(Connectedness)

Bazı graflar tek bir parça halinde iken diğerleri çeşitli parçalardan oluşuyor olabilir. Bu fikri daha belirgin hale getirmek için yolları kullanabiliriz. Eğer bir graf çeşitli şehirleri bağlayan yol ağını temsil ediyorsa aklımıza şu soru gelebilir: her yoldan bir kere geçerek ve her şehre sadece bir kez uğrayarak aynı şehirden başlayıp aynı şehirde biten bir yolculuk yapılabilir mi?

Tanım: (i) G grafında n uzunluğunda ayrıt dizisi; i =1, 2, … n-1 için e_i ve e_{i+1} komşu olmak üzere e_1, e_2,…e_n ayrıtlarının dizisidir. Ayrıt dizisi, $\delta(e_i)=\{v_{i-1}, v_i\}$ olmak üzere $v_0,v_1,v_2,\dots v_{n-1},v_n$ düğüm dizisini belirler. v_0'a ilk düğüm, v_n 'e son düğüm denir.

(ii) Bir **yol (path)** tüm ayrıtları birbirinden ayrı (distinct) olan ayrıt dizisidir. Buna ek olarak eğer tüm düğümler de birbirinden ayrı ise bu yol basit (simple) yoldur.

Diğer bir tanım(**Yol**): Bir grafta, v_1 ve v_r düğümü arasındaki bir yol, düğümlerin $v_1,e_1,v_2,e_2,v_3,e_3,\dots,e_r,v_r$ şeklindeki ayrıtların sonlu bir dizisidir. Burada, e_k, v_{k-1} ve v_k düğümleri arasındaki ayrıttır.

(iii) v_0=v_n ise ayrıt dizisi kapalıdır(closed). En az bir ayrıt içeren basit kapalı bir yol devre (circuit) olarak adlandırılır.

Bir ayrıt dizisi grafın diyagramında kalemi kâğıdın üzerinden kaldırmadan çizebileceğimiz herhangi sonlu ayrıt dizisidir. Ayrıtlar tekrar edilebilir veya çevrimler tekrarlanabilir. Ayrıt dizileri çok genel olduklarından kullanıma uygun değillerdir ve bu yüzden yollar tanımlanmıştır. Bir yolda aynı ayrıttan birden fazla geçmeye izin verilmez. Buna ek olarak eğer aynı düğümü birden fazla ziyaret etmiyorsak bu yol basit yoldur. Ayrıt dizisi veya yol, bir yerden başlayıp aynı yerde bitiyorsa kapalıdır.

Herhangi bir graf doğal olarak bileşen (component) adı verilen belli sayıda bağlı alt graflara bölünebilir. Bileşenler maksimal bağlı alt graflar olarak tanımlanabilirler. Bunun anlamı G_1, G ‘nin bağlı bir alt grafı ise ve kendisi G ‘nin başka herhangi bir bağlı alt grafının alt grafı değilse, G ‘nın bileşenidir. Bu ikinci durum maksimal terimi ile anlatmak istediğimiz şeydir yani Σ, $G_1 \leq \Sigma$ olacak şekilde bir bağlı alt grafsa $\Sigma = G_1$, böylece G ’in G_1 ‘den daha büyük bağlı bir alt grafı yoktur

G şekil 7.1 ‘de gösterilen graf olsun. Bu grafın komşuluk matrisi:

$$A=\begin{vmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}. \text{ Dir.}$$

A’ nın (i,j). elemanı v_i ve v_j düğümlerini bağlayan ayrıtların sayısıdır. Bunu bu iki düğümü bağlayan 1 uzunluğunda ayrıt dizilerinin sayısı şeklinde düşünebiliriz. Bu durumda, komşuluk matrisinin karesi:

$$A^2=\begin{vmatrix} 3 & 3 & 3 & 0 \\ 3 & 5 & 1 & 0 \\ 3 & 1 & 5 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}. \text{dir.}$$

A^2’ de (i,j). eleman v_i ve v_j ‘yi bağlayan 2 uzunluğunda ayrıt dizilerinin sayısını temsil eder. Örneğin, (2,2). eleman 5’ tir ve v_2 ‘yi kendisine bağlayan 2 uzunluğunda 5 tane ayrıt dizisi vardır: e_2, e_2; e_4, e_4; e_5, e_5; e_4, e_5; e_5, e_4.

Bunun niçin böyle olduğunu görmek zor değildir. A^2’ nin (i,j). elemanı A’ nın i. satırı ile j. sütununun çarpılması ile elde edilir.

$$a_{ij} = \sum_{k=1}^{n} a_{ik} a_{kj} .$$

Toplamdaki r. terim $a_{ir}a_{rj}$, v_i ve v_r ‘yi bağlayan ayrıtların sayısı ile v_r ve v_j ‘yi bağlayan ayrıtların sayısının çarpımıdır. Bir başka ifade ile v_i ve v_j ‘yi v_r aracılığı ile bağlayan 2 uzunluğundaki ayrıt dizilerinin sayısıdır. Tüm k değerleri için ortaya çıkanların toplanması v_i ve v_j ‘yi bağlayan 2 uzunluğunda ayrıt dizilerinin sayısını verir.

Benzer şekilde A^3 ‘te (i,j). eleman v_i ve v_j ‘yi bağlayan 3 uzunluğundaki ayrıt dizilerinin sayısıdır. Bu graf için

$$A^3=\begin{vmatrix} 9 & 9 & 9 & 0 \\ 9 & 5 & 13 & 0 \\ 9 & 13 & 5 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}.$$

Teorem 7.2: G, düğüm kümesi $\{v_1, v_2, \ldots, v_n\}$ ve komşuluk matrisi A olan bir graf olsun. A^n ‘ in (i,j). elemanı v_i ve v_j ‘ yi bağlayan n uzunluğunda ayrıt dizilerinin sayısıdır.

Yönlü Graflarda yol

Bir yönlü grafta v düğümünden w düğümüne olan yönlü bir yol, düğümlerin ve yayların $v_1,a_1,v_2,a_2,v_3,a_3,\ldots,v_r,a_r,v_{r+1}$ şeklindeki sonlu bir dizisidir. Burada, ilk düğüm v ve son düğüm w ve a_i, v_i den v_{i+1}. düğüme olan yaydır. Eğer v’den w’ye bir yönlü yol var ise, v, w’ye bağlıdır ve w, v’den bağlıdır. Bir düğümden kendine olan bir yönlü yol, bir kapalı yönlü yoldur. Eğer düğümlerin çifti birbirine bağlı ise bu düğümlere **kuvvetli bağlı** çift denir. Bir grafta her bir düğüm çifti kuvvetli bağlı ise bu graf **kuvvetli bağlı**dır. Aksi halde **zayıf bağlı** graftır.

Eğer {v,w} kuvvetli bağlı çift ise, vRw ile, V düğüm kümesini iki ayrık alt küme sınıfına ayıran bir eşdeğerlik bağıntısı tanımlanır. Bu alt kümelerin her birine yönlü grafın bir kuvvetli parçası denir.

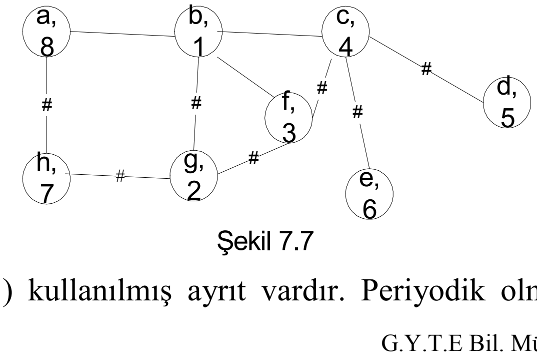
Teorem 7.3 : Eğer A bir yönlü grafın komşuluk matrisi ise, A’nın k. kuvvetinin($k \geq 1$) (i,j) elemanı, i den j’ye olan k- yönlü yolun sayısını verir.

Grafların Bağlılık Testi

Tanım: Bir grafta eğer birbirinden ayrı düğümlerini bağlayan bir yol varsa bağlıdır(connected).

Verilen bir grafın bağlı olup olmadığının sorulması doğaldır. Elbette grafın şemasından bağlı olup olmadığının görülmesi kolaydır. Ancak büyük graflarda bu yöntem makul değildir. Grafın bilgisayara girilmesi durumunda, bağlılık testi için bir algoritma gereklidir. Böyle bir algoritma, grafın düğümlerinin yeniden etiketlendiği ilk derinlik arama(dept-first search) tekniğidir.

G grafının düğümleri $v_1,v_2,\ldots,v_n$ olsun.
Keyfi bir nokta seç ve onu 1 olarak etiketle.
1’e komşu etiketsiz bir düğüm seç ve onu 2 olarak etiketle.
{1,2} yi kullanılan ayrıt olarak işaretle ve tekrar kullanma.
Benzer şekilde v_i düğümünü k ile etiketle. Bu düğüme komşu olan ve etiketsiz olan bütün düğümleri ara ve birisini seçerek (k+1) olarak etiketle.
{k,k+1} i kullanılmış kenar olarak işaretle. Şimdi k’nın bütün komşu düğümleri etiketlenmiş olabilir.
Eğer öyle ise, (k-1). düğüme git ve onun etiketsiz komşu düğümlerini ara. Eğer böyle bir düğüm var ise onu (k+1) olarak etiketle ve {k-1,k+1} kenarı kullanılmış kenar olarak işaretle.
İşleme bütün düğümler işaretleninceye kadar devam et veya en az bir etiketlenmemiş düğüm ile 1. düğüme dön.



Şekil 7.7

Örnek: Şekil 7.7’de 8 düğümlü {a,b,c,d,e,f,g,h} olan grafta, b seçilerek 1 olarak etiketlemiş ve {1,2} kullanılmış ayrıt olarak işaretlenerek devam edilmiş ve diğer düğümler etiketlenmiştir.

İlk durumda graf bağlıdır ve tam olarak (n-1) kullanılmış ayrıt vardır. Periyodik olmayan bir graf n düğüm içerir ve bu (n-1) ayrıta grafın ilk derinlik arama uzaklık ağacı(first-depth search spanning tree) denir. Eğer DFS tekniği ile tüm n adet düğüm etiketlenmemiş ise graf bağlı değildir sonucuna varırız. Bu algoritmanın en kötü durumda karmaşıklığı, eğer m ayrıt var ise en fazla 2m araştırma yapılacak ve n adet etiketlenecek düğüm olacaktır. Böylece karmaşıklık en kötü durumda n+2m olacaktır. m'nin en büyük değeri n(n-1)/2 olduğundan(bütün düğüm çiftleri arasında bir ayrıt olduğu durum) en kötü durumda karmaşıklık $O(n^2)$ olacaktır.

8.2 Yollar ve Devreler

Euler Yolları (Eulerian Paths)

Tanım: G grafındaki bir Euler Yolu, G' nın tüm ayrıtlarını kenar olarak bir kere içeren kapalı bir yoldur. Kapalı bir Euler yolu bir Euler devresidir. Bir graf içinde en az bir Euler Yolu barındırıyorsa bu graf Euler grafıdır.

Euler devresi fikri meşhur Königsberg Köprüsü Probleminden ortaya çıkmıştır. Pregel nehri Königsberg kasabasının içinden akmaktadır. Nehrin ortasında şekil 7.4 (a) daki gibi nehrin kıyılarına ve birbirine köprüler ile bağlı iki ada bulunmaktadır. Königsberg kasabasının vatandaşları için problem, kıyıların veya adaların birinden başlayıp tüm köprülerden sadece bir kez geçerek başladığımız yere yürüyebilir miyiz?

Euler, öncelikle şekil 7.4 (b) ' deki gibi Königsberg coğrafyasının gerekli özelliklerini bir graf ile gösterdi. Her bir nehir kıyısı ve adalar bir düğüm ile köprüler de ayrıtlar ile temsil edildi. Graf teorisi terimleri ile problem şu hale geldi: grafın tüm ayrıtlarını içeren kapalı bir yol var mıdır?

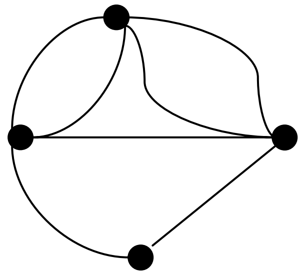
Bir yolda hiçbir ayrıt dan birden fazla geçilemeyeceğinden Euler yolu tüm ayrıtları sadece bir kez içerir fakat düğümlerden birden fazla geçilebilir.

Bağlı bir G grafında Euler yolu olup olmadığını belirlemek için gereken durumu tanımlamak çok kolaydır: bütün düğümlerin derecesi çift olmalıdır.Bunu görmek için G bağlı ve Euler yoluna sahip olsun. G bağlı olduğundan Euler yolunun düğüm dizisi bütün düğümleri içerir. Yol ne zaman bir düğümden geçse bu derecesine iki katkı yapar. Tüm ayrıtlar yolda bir kere bulunduğundan her düğüm çift dereceye sahip olmalıdır.

Königsberg'dekiler aradıkları yolu bulamamakta haklıdırlar zira böyle bir yol yoktur. Problemi temsil eden şekil 7.4 (b)' deki graf bağlıdır fakat gerekli koşulu sağlamaz. Aslında tüm düğümlerin derecesi tektir. Aşağıdaki teorem bu soruyu sabitler.

Teorem 7.4: Çevrim içermeyen bağlı bir G grafı sadece ve sadece tüm düğümleri çift dereceli ise Euler grafıdır.

Örnek: Şekil'deki grafta tüm düğümlerin dereceleri çift olduğu için bu bir Euler grafıdır.



Bir grafta Euler devresi bulmak için kolay bir yol Fleury'in algoritması olarak bilinir. Bu yöntemde, herhangi bir düğümden başlanır ve geçilen bir ayrıt silinir. Aynı zamanda, yardım etseniz bile bir köprü asla geçilmez. Eğer geçilen ayrıtlar silinerek başlanılan noktaya ulaşılabiliyor ise, devre Euler devresidir ve graf Euler dir.

Teorem 7.5: Bağlı ve çevrim içermeyen Euler olmayan bir G grafında ancak ve ancak tam olarak iki tek dereceli düğüm var ise bir Euler yolu vardır.

İspat: Eğer G, u'dan v'ye bir Euler yoluna sahip ise, u ve v'nin her ikisi de tek dereceli ve bu yolda her düğümden geçip her ayrıt bir kez ziyaret edildiği için diğer düğümlerin her biri çift dereceli olmalıdır. Diğer taraftan, G nin u ve v olan iki tek dereceli düğüm ile bağlandığını kabul edelim. Bu u ve v düğümleri ya komşudur veya değildir. İlk durumda ikisi arasında bir e ayrıtı olsun. Her bir düğümü çift dereceli olan G' grafını elde etmek için e'yi silelim.Eğer G' bağlı ise u'dan başlayıp v'ye kavuşan bir Euler yolu elde edilir. Eğer G' iki parça ise, birisi u'yı içeren G_1 diğeri de v'yi içeren G_2'dir. Elbette her ikisi de Euler grafıdır. Dolayısı ile bir Euler yolu bulunabilir.

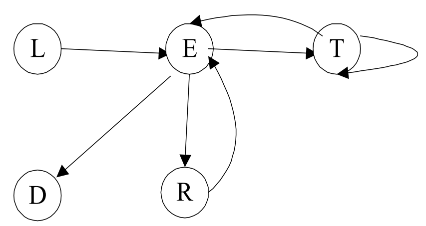
Teorem 7.6: Bir zayıf bağlı yönlü graf ancak ve ancak her bir düğümün giriş ve çıkış dereceleri aynı ise bir yönlü Euler devresi içerir.

Kodlama ve de Brujin yönlü grafları

Euler yol ve devrelerinin, bilgisayar bilimleri, yöneylem araştırması, kriptografi ve taşıma problemleri gibi ilginç ve yararlı uygulamaları vardır. Bu bölümde birkaç örnek verilecektir.

Çinli postacı problemi, keyfi bağlı bir ağın Euler grafına genişletildiği bir optimizasyon problemidir. Problemde; Bir posta taşıyıcısı postaneden çıkar, bölgesindeki her bir bloğa postaları dağıtır ve ofisine geri döner. Eğer yolu üzerindeki her bir cadde köşesini bir düğüm ve iki köşe arasındaki yolu bir ayrıt olarak alırsak, bu problemin modeli olan bir graf elde ederiz. Eğer graf bir Euler Grafı ise postacı her bir caddeyi bir kere geçmelidir. Eğer Euler grafı değil ise, postacı bazı caddeleri tekrarlayacaktır. Bir optimizasyon problemi bu tekrarlanan caddeleri toplam gidilen yolu en aza indirecek şekilde konumlandırmaktır.

Diğer bir problemde Euler grafının kodlama teorisindeki uygulamasıdır. m harf uzunluğunda ve n farklı harfi olan bir kelime bir zayıf bağlı, n düğümlü, m-1 yaylı G grafı olarak ilişkilendirilebilir. Burada, eğer kelimenin ilk harfi ve son harfi farklı ise yönlü bir Euler yolu, aynı ise yönlü bir Euler devresinin oluştuğu görülür. Uygulama olarak LETTERED kelimesi ile oluşturulan yönlü Euler grafında m=8 ve n=5'dir. Şekil 7.8'de gösterilen yönlü grafta,ilk harf L 'den son harf D'ye gitmek için 5 düğüm ve 8 yaydan geçilmelidir.



Şekil 7.8

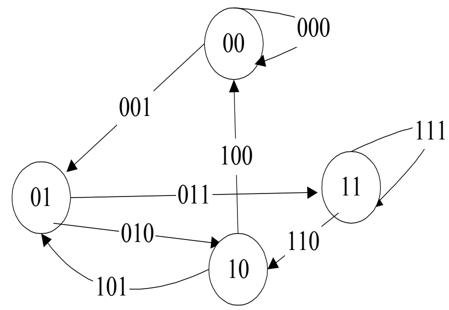
$A_1, A_2, \ldots A_n$ gibi n farklı harfli bir kelimde, $f(A_i)$, A_i'nin kelimdeki frekansı olsun. Buradan n harfin frekanslarının toplamı m dir. Yönlü grafta, A_i'den A_j'ye olan yay sayısını gösteren m_{ij}, A_j 'nin A_i'nin hemen arkasından gelme sayısını göstersin. Örneğim MATHEMATICS kelimesinde, $m_{AT}=2$, $m_{TA}=0$, $m_{TH}=1$ ve böyle devam eder. Buradan $M=(m_{ij})$, nxn boyutlu bir matris olarak tanımlanır. M'de i. Satırın satır toplamı A_i düğümünün çıkış derecesi, j. kolonun kolon toplamı A_j'nin giriş derecesidir.

Böylece, n farklı harfli bir kelimeye karşılık olarak n pozitif tamsayının frekansı ve elemanları pozitif olan nxn'lik bir matrisimiz vardır. Örneğin "LETTERED" kelimesinin frekans kümesi, {1,3,1,1,2} ve 5x5 lik matris yanda gösterilmiştir.

$$
M = \begin{array}{cccccccc}
 & D & E & L & R & T & \\
D & 0 & 0 & 0 & 0 & 0 & 0 \\
E & 1 & 0 & 0 & 1 & 1 & 3 \\
L & 0 & 1 & 0 & 0 & 0 & 1 \\
R & 0 & 1 & 0 & 0 & 0 & 1 \\
T & 0 & 1 & 0 & 0 & 1 & 2 \\
 & 1 & 3 & 0 & 1 & 2 &
\end{array}
$$

Euler graflarının diğer bir uygulaması da de Brujin graflarıdır. Uzunluğu n-1 olan 2^{n-1} ikili kelime var. Burada 2^{n-1} düğümlü bir yönlü graf oluşturacağız. Her bir n-1 uzunluğundaki kelime bir düğümde olsun. Her bir $v=a_1a_2\ldots a_n$ şeklindeki düğümden , $a_2a_3\ldots a_n0$ ve $a_2a_3\ldots a_n1$ şeklinde iki n harfli kelimeyi temsil edecek şekilde v_1 ve v_2 yaylarını çiz. Böylece, n uzunluğundaki kelimeleri temsil eden 2^n yaylı yönlü graf çizilmiş olur. Bu graf G(2,n) zayıf bağlı de Brujin grafıdır ve her bir düğümün giriş ve çıkış derecesi eşit olduğundan Euler grafıdır. G(2,3) yönlü grafı Şekil 7.9'da gösterilmiştir.

Daha genel olarak, p harfli alfabe için G(p,n) giriş ve çıkış derecesi p olan ve p^{n-1} düğüm p^n yay içeren bir de Brujin yönlü grafıdır. Böylece G(p,n) bir Euler grafıdır. Bu yönlü grafta bir Euler yolu, bir dizide p^n yay içerir. Bu kelimelerin ilk harfleri ile bir dizi oluşturmak istenirse, böyle bir dizi, $a_1a_2\ldots a_r$ dir, burada $r=p^n$ dir. Buradan uzunluğu n olan r farklı kelime $a_ia_{i+1}\ldots a_{i+n-1}$ şeklindedir, burada alt indiste belirtilen toplama işlemi modulo r şeklindedir. Örneğin p=2, n=3 ise a_9 a_1 ile aynı olacaktır. Örneğin, Şekil 7.9'daki yönlü Euler grafında, 00'dan başlayan devre, 000,001,011,111,110,101,010,100 olan sekiz yay dizisini içerir. Bunların ilk harfleri ile oluşturulan dizi, 00011101 şeklinde ve buradan oluşturulacak $a_ia_{i+1}a_{i+2}$ şeklindeki üç harflik dizi de $a_7a_8a_9=a_7a_8a_1=010$ olacaktır.



Şekil 7.9

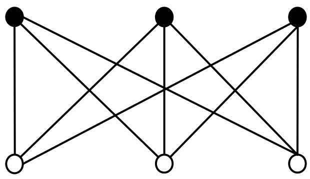
Buradan de Brujin dizisini biçimsel olarak, p ve n iki pozitif tamsayı için tanımlarsak; Eğer S, p harf içeren bir alfabe ise, r ($r=p^n$) harfin $a_1a_2\ldots a_r$ dizisine B(p,n) ile gösterilen de Brujin Dizisi denir. S'den n uzunluğundaki her bir kelime, $a_ia_{i+1}\ldots a_{i+n-1}$ (i=1,2,,…,r) olarak gerçeklenebilir. Burada alt indisteki toplama işlemi modulo r olarak gerçeklenir.

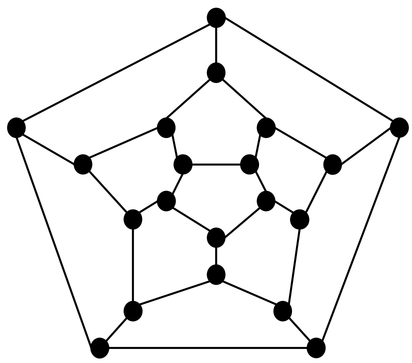
Hamilton Devreleri (Hamiltonian Circuits)

Benzer bir problem de herhangi bir ayrıtdan birden fazla geçmemek kaydıyla her bir düğümü sadece bir kez ziyaret edip başladığımız yere geri dönebilir miyiz? Şeklinde sorulabilir. Bu problem Hamilton tarafından irdelenmiştir ve ismi bu yollar ile birlikte anılmaktadır.

Tanım: Eğer bir grafta her bir düğümden sadece bir kere geçilen bir yol varsa iki düğüm arasındaki yola, **Hamilton yolu** denir. Bir graftaki Hamilton devresi tüm düğümlerden bir kez geçen bir devredir. Bir graf içinde bir Hamilton devresi barındırıyorsa Hamilton grafıdır. Her bir düğümden tam olarak bir kere geçen ve tüm ayrıtların farklı olduğu bir kapalı yol Hamilton devresidir.Bir graf Hamilton devresi içeriyorsa bu bir Hamilton grafıdır. Bir yönlü grafta, bir düğümden diğerine geçen yönlü yol eğer her düğümden bir kere geçerse bu yönlü Hamilton yoludur. Bir kapalı yönlü Hamilton yolu bir yönlü Hamilton devresidir.

Örnek 7.3: Şekil 7.10 'da iki tane Hamilton devresi vardır.





Şekil 7.10 (a) Şekil 7.10 (b)

Euler grafları basit bir karaktere sahipken aynı durum Hamilton grafları için doğru değildir.

Aslında bir asırdan beri üzerinde çalışıldığı halde Hamilton graflarının karakteri hakkında her şey bilinmemektedir (Karakter ile bir grafın Hamilton olması için gerek ve yeter koşul kastedilmiştir). Bu graf teorisinin çözülememiş büyük problemlerinden biridir. Açık bir gerek koşul grafın bağlı olmasıdır. Ayrıca çeşitli yeter koşullar da bilinmektedir.

Bununla birlikte Hamilton grafı için aşağıdaki teoremler verilebilir.

Teorem 7.7: Bir n düğümlü($n\geq 3$) basit grafta, eğer komşu olmayan düğümlerin her çiftinin derecesi toplamı en az n ise bu bir Hamilton grafıdır.

Sonuç: Eğer, G n ($n\geq 3$) düğümlü basit bağlı bir graf ise ve tüm v düğümleri için derecesi $\sigma(v)\geq n/2$ ise G Hamilton'dur.

Dereceler ile ilgili koşul G 'nin Hamilton olması için gerek koşul değildir o halde, bu koşulu sağlamayan bir graf da Hamilton olabilir. Şekil 7.10 (b) 'ye bakarak bunu görebiliriz. Grafın 15 düğümü vardır, her düğümün derecesi 3 'tür fakat hala Hamilton grafıdır.

Teorem 7.8: Bir n düğümlü basit grafta, eğer komşu olmayan düğümlerin her çiftinin derecesi toplamı en az n-1 ise bu graf bir Hamilton yolu içerir.

Sonuç: Bir n düğümlü basit grafta, eğer her bir düğümün derecesi en az (n-1)/2 ise bu graf bir Hamilton yolu içerir.

Hamilton devresinin Uygulaması: Hamilton yolu ve devrelerinin ilginç uygulamaları vardır. Buna bir örnek aşağıda verilmiştir.

Örnek(Satıcı Seyahat Problemi): Bir ülkedeki şehirler düğümleri gösterecek, uçak seferi olan şehirler arasındaki bağlantılarda ayrıtlar olmak üzere bir graf oluşturulsun. Bir satıcı her bir şehre bir kere uğrayıp tekrar başladığı yere dönmek üzere bir yolculuk programı yapmak istiyor. Böyle herhangi bir tur bir Hamilton devresidir. Böyle bir devrenin var olduğu kabul edilirse, toplam maliyeti en az olan bir yol bulmak bir optimizasyon problemidir.

Örnek(Planlama) : Bir makine atölyesinde n adet makine bulunsun. Bir iş keyfi bir sırada olmayacak şekilde bu makineler arasında yapılacaktır. Her bir makine bir yönlü grafta bir düğümü temsil etsin. Her bir düğümden diğerine bir yay çiz. Bu yönlü grafta herhangi bir yönlü Hamilton yolunun bulunması bir planlamadır. Eğer, bir işin i.makineden j. Makineye giderken gerekli olan düzenleme zamanı c_{ij} ise, en az zamana sahip bir planlamayı bulmak yine bir optimizasyon problemidir.

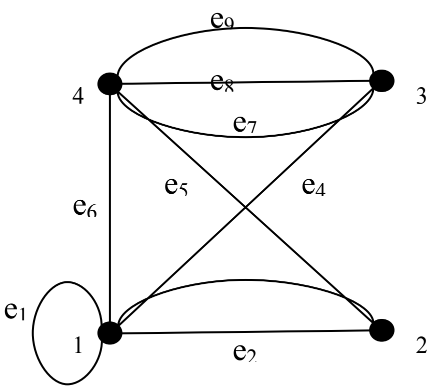
8.3 Grafların İzomorfizmi

Aşağıdaki gibi tanımlanan G ve Σ graflarını düşünelim. G' nin düğüm kümesi {1,2,3,4}, komşuluk matrisi A ve Σ 'nin düğüm kümesi {a,b,c,d}, komşuluk matrisi B olsun.

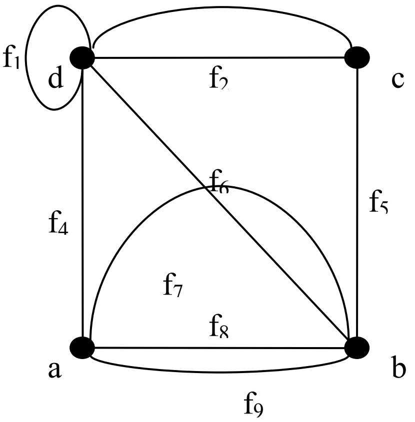
Aşağıdaki gibi tanımlanan G ve Σ graflarını düşünelim. G' nin düğüm kümesi {1,2,3,4}, komşuluk matrisi A ve Σ 'nin düğüm kümesi {a,b,c,d}, komşuluk matrisi B olsun.

$$A = \begin{vmatrix} 1 & 2 & 1 & 1 \\ 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 3 \\ 1 & 1 & 3 & 0 \end{vmatrix} \qquad B = \begin{vmatrix} 0 & 3 & 0 & 1 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 1 & 1 & 2 & 1 \end{vmatrix}$$

G ve Σ graflarını temsil eden diyagramlar şekil 7.11 ‘da gösterilmiştir.



Şekil 7.11 (a)



Şekil 7.11 (b)

Biraz dikkatli bakılırsa şekil 7.11‘ de gösterilen grafların aynı olduğu görülebilir. Σ grafındaki a,b,c,d düğümlerini 3,4,2,1 şeklinde; ve f_i ayrıtlarını i=1,…,8 için e_i ile tekrar etiketlersek şekil 7.11 ‘deki iki diyagrama aynı grafın farklı gösterimleri şeklinde bakabiliriz. Tabii ki, G ve Σ grafları birebir aynı değildir. Örneğin farklı düğüm kümelerine sahiptirler. Öte yandan aynı yapıya sahiptirler. G ve Σ grafları izomorfiktir graflar diyebiliriz.

Σ ‘nın düğümlerini yeniden etiketleyerek G ve Σ ‘nın düğüm kümeleri arasında bir bijeksiyon tanımlamış oluruz.

Tanım: G ve Σ iki graf olsun. G ‘den Σ’ a bir izomorfizm (Θ, Φ) bir bijeksiyon çiftinden oluşur.

$$\Theta: V_G \rightarrow V_\Sigma \text{ ve } \Phi: E_G \rightarrow E_\Sigma$$

öyle ki G’ nın tüm e ayrıtları için eğer $\delta_G(e) = \{v, w\}$ ise $\delta_\Sigma(\Phi(e))=\{ \Theta(v), \Theta(w)\}$.

İki graf, bir graftan diğerine bir izomorfizm varsa izomorfiktir denir ve G Σ şeklinde gösteririz.

$\delta_G(e) = \{v, w\}$ ise $\delta_\Sigma(\Phi(e))=\{ \Theta(v), \Theta(w)\}$ olması şartının anlamı iki grafın ayrıtları ve düğümleri arasındaki uyuşmanın doğru şekilde sağlandığından emin olmak içindir.

Basit bir G grafı için G’ dan Σ ‘ya bir izomorfizm tanımlamak için sadece uygun $\Theta: V_G \rightarrow V_\Sigma$ düğüm bijeksiyonunu belirlemek gerekir. Bunun nedeni herhangi düğüm çiftini birleştiren en az bir tane ayrıt vardır o halde, bir kez Θ tanımlandığında gerekli özellikleri sağlayan sadece bir tane $\Phi: E_G \rightarrow E_\Sigma$ fonksiyonu vardır.

İzomorfik grafların aynı yapıya sahip olmaları gerektiğinden birine ait graf teorisine dahil herhangi bir özellik diğerinde de bulunmalıdır. Bu özelliklerin bir kısmı aşağıdaki teoremde sıralanmıştır.

Teorem 7.8: (Θ, Φ) G’ dan Σ ‘ya bir izomorfizm olsun. Bu durumda;
(i) G ve Σ aynı sayıda düğüme sahiptir;

(ii) G ve Σ aynı sayıda ayrıt’ e sahiptir;

(iii) G ve Σ aynı sayıda bileşene sahiptir;

(iv) birbirine karşılık gelen düğümler aynı dereceye sahiptir;

(v) G basitse, Σ da öyledir;

(vi) G Euler grafı ise Σ da Eulerdir.

(vii) G Hamilton grafı ise Σ da Hamiltondur.

İzomorfizm Prensibi

İki grafın izomorfik olduğunu göstermek için birinden diğerine bir izomorfizm bulunmalıdır; iki grafın izomorfik olmadığını göstermek için ise bir grafın sahip olduğu ama diğerinin sahip olmadığı bir graf teorisine dahil bir özellik bulunmalıdır.

8.4 Düğüm Boyama, Ağaçlar ve Düzlemsel Graflar

Eğer bir grafta, iki komşu düğüm aynı renkte olmayacak şekilde, her bir düğüme bir renk verilirse graf boyalıdır denir. Eğer böyle bir boyama en çok k renk kullanılarak mümkün olursa, graf k-renklidir. Böyle k-renkli bir G grafında en küçük k değeri G'nin kromatik sayısıdır.

Bir grafta ancak ve ancak hiç ayrıt yok ise kromatik sayısı birdir. n düğümlü bir tam grafın kromatik sayısı n, iki parçalı grafın kromatik sayısı ise 2 dir. Bir ağacın kromatik sayısı 2 dir. p düğümlü bir devre ancak ve ancak, p çift ise 2-renkli, benzer şekilde eğer bir G grafı tek devre(devredeki ayrıt sayısı tek) içeriyorsa, G grafı 2-renkli değildir. Eğer bir grafta hiç tek devre yoksa graf, 2-renklidir.

Graflarda düğüm boyama için değişik algoritmalar geliştirilmiştir. İki örnek algoritma aşağıda verilmiştir.

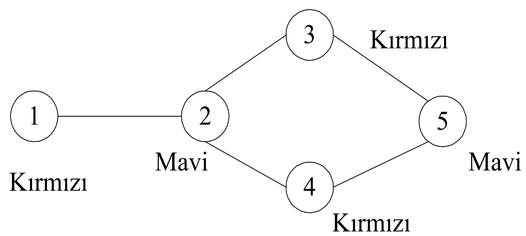
Graf Boyama için açgözlü bir algoritma

1. Bir düğümü al ve kullanılmayan bir rengi ver

2.Komşu olan düğümlere farklı komşu olmayanlara mümkün olduğunca aynı renk ver.

3. İşlemleri bütün düğümler için tekrarla.

Şekil 7.12'de gösterilen graf için boyama örneği verilmiştir. İlk 1 nolu düğümden başlayıp kırmızı renk, ona komşu olmayan 3 ve 4 nolu düğümler yine kırmızı, birbirine komşu olmayan 2 ve 4 nolu düğümlere ise mavi renk verildi. Böylece sadece iki renk kullanılmış olur.



Şekil 7.12

Diğer bir algoritma: Yukarıdaki basit algoritmaya biraz sezgisel yaklaşım ekleyerek geliştirelim.

1. Düğümleri($v_1,v_2,\ldots,v_n$) derecelerine göre azalan sırada sırala $\sigma(v_1)\geq \sigma(v_2)\geq\ldots\geq \sigma(v_n)$
2. Renk 1'i v_1'e ve listede v_1'e komşu olmayan sonraki düğüme(eğer var ise) ver
3. Renk 2'yi listede boyanmayan ve renk 2 ile boyanmış düğümlere komşu olmayan düğümlere ver.
4. Eğer boyanmayan düğüm kalmış ise, renk 2 yi ver.
5. Bu işleme bütün düğümler boyanıncaya kadar devam et.

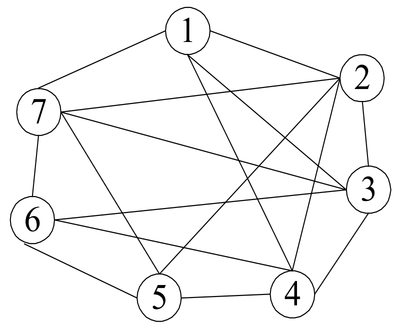
Graf boyama fikri birçok planlama probleminin çözümünde faydalıdır.

Örnek: Bir üniversitede final sınavlarının çakışmayacak şekilde planlamasının yapılması:

Çözüm: Bu planlama problemi, düğümler dersleri ve düğümler arasındaki ayrıtlar eğer derslerde ortak öğrenci var ise onu temsil eden bir graf modeli oluşturulur. Final sınavı için her zaman dilimi farklı bir renk ile gösterilir. Sınav planlaması oluşturulan grafın boyanması problemine dönüşür. Örneğin, 7 adet planlanacak sınav olsun. Dersler 1-7 arasında numaralandırılmış olsun. 1 ve 2, 1 ve 3, 1 ve 4, 2 ve 3, 2 ve 4, 2 ve 5, 2 ve 7, 3 ve 4, 3 ve 6 , 3 ve 7, 4 ve 5, 4 ve 6, 5 ve 6, 5 ve 7, 6 ve 7 nolu dersler ortak öğrenci bulundursunlar. Şekil 7.13'de gösterilen grafta, planlama problemi düğüm boyama problemine dönüşmüş olur.

Bu grafın kromatik sayısı 4 olduğundan sınavlar için 4 zaman dilimi gereklidir. Graftaki renkler, 1ve 6 kırmızı, 2 Mavi ,3 ve 5 Yeşil, 4 ve 7 kahverengi olarak boyanır. Bu zaman dilimlerindeki sınavlar aşağıdaki tabloda gösterilmiştir.

Zaman Dilimi	Dersler
I	1,6
II	2
III	3,5
IV	4,7

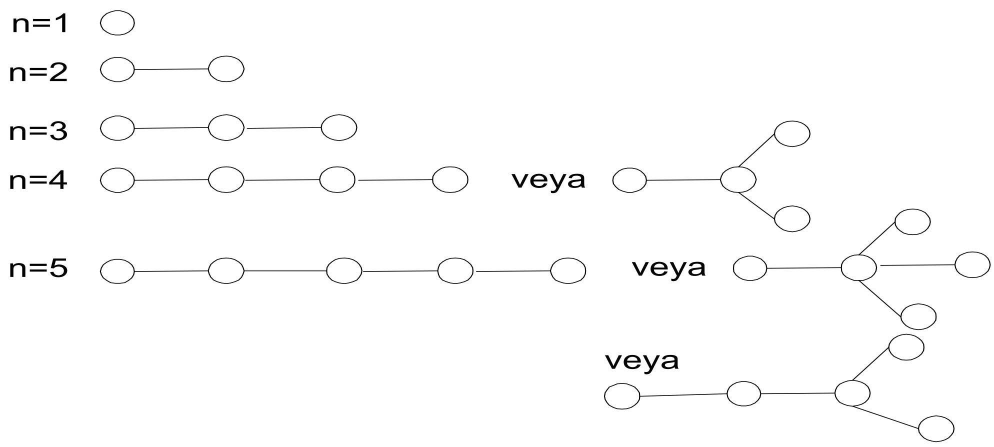


Şekil 7.13

Ağaçlar

Tanım: İçinde devre (circuit) içermeyen bağlı graflara **ağaç (tree)** denir.

Tanımdan da açıkça görüldüğü gibi bir ağaçta loop veya çoklu ayrıt yoktur. Herhangi bir loop kendi başına bir devredir ve e_i ve e_j aynı düğüm çiftinin bağlıyorsa e_i, e_j dizisi de bir devredir. Bazı ağaç örnekleri Şekli 7.14'de gösterilmiştir..



Şekil 7.14. Düğüm sayısına göre farklı ağaçlar.

Teorem 7.9.: n düğümlü bir G grafında aşağıdakiler eşdeğerdir.

i. G bir ağaçtır.
ii. G'de her düğüm çifti arasında, sadece bir yol vardır.
iii. G bağlıdır ve G'deki her bir ayrıt bir köprüdür.(köprü silinince graf bağlı olmaktan çıkar)
iv. G bağlıdır ve (n-1) ayrıtı vardır.
v. G çevrimsizdir ve (n-1) ayrıtı vardır.
vi. G çevrimsizdir ve, G'de komşu olmayan iki keyfi düğüm bir ayrıt ile birleştirildiği zaman sonuçtaki genişleyen G' grafı tek bir tur içerir.

vii. G bağlıdır, ve eğer G'de herhangi komşu olmayan iki keyfi düğüm bir ayrıt ile birleştirilirse, elde edilen yeni grafın tek bir çevrimi vardır.

Genel olarak, ağaçlar ile ilgili algoritmalar üç türlüdür.

- Verilen bir ağaçta arama ve etiketleme algoritmaları
- Farklı türlerde ağaç oluşturmak için algoritmalar.
- Özel bir türdeki ağaçları saymak için algoritmalar.

Ağaçlar ile ilgili Tanımlar ve özellikleri

- Bir ağaç, çevrim içermeyen bir bağlı yönsüz graftır.
- Bir yönsüz graf ancak ve ancak, herhangi iki düğümü arasında tek bir basit yol var ise bir ağaçtır.
- Bir köklü ağaç, bir ağaçtan bir düğümün kök olarak belirlenmesi ve herbir ayrıt kökten yönlendirilerek elde edilen bir yönlü graftır.
- Bir köklü T ağacında, (u,v) bir yönlü ayrıt olsun,
  - ✓ u, v'nin ebeveyni ve v'de u'nun çocuğudur,
  - ✓ aynı ebeveyne sahip çocuklara kardeş denir;
  - ✓ bir v düğümünün kök haricindeki ataları, kökten v'ye kadar olan yol üzerindeki düğümlerdir,
  - ✓ v düğümünün torunları v'yi ata olarak gören düğümlerdir;
  - ✓ bir yaprak, hiç çocuğu olmayan bir düğümdür,
  - ✓ çocuğu olan düğümlere iç düğümler denir;
  - ✓ torunlarıyla, birlikte bir v düğümü ve bu torunlara komşu bütün ayrıtlar bir alt graf oluşturur.
- Her iç düğümü ≤ *m* çocuğa sahip olan bir köklü ağaca m-ilişkili ağaç denir, eğer m=2 ise ikili ağaçtır.
- Bir köklü ağaçta, bir v düğümünün seviyesi, kökten v'ye olan tek yolun uzunluğudur.
- Bir köklü ağacın yüksekliği, düğümlerin seviylerinin en büyüğüdür.
- Yüksekliği h olan bir köklü m-ilişkili ağaç, eğer bütün yapraklar *h* veya *h-1* seviyesinde ise dengeli ağaçtır.
- Bir sıralı köklü ağaçta, herbir iç düğümün çocukları sıralıdır. Eğer bir düğümün iki çocuğu varsa, ilk çocuğa sol alt ağaç, ve sağ çocuğa sağ alt ağaç denir.
- Ağaçlar; doymuş hidrokarbonları, Kuruluşları, Dosya kataloglarını, parallel işlem için ağ iç bağlantılarını modellemek için kullanılabilir.
- ***Ağaçların Özellikleri***
  - ▪ n düğümlü bir ağacın tam olarak n-1 ayrıtı vardır.
  - ▪ i adet iç düğümü olan bir tam m ilişkili ağaçta n=m.i +1 düğüm bulunur.
  - ▪ Bir tam m-ilişkili;
    - ❑ *n düğümlü* ağacın, $i = (n-1)/m$ iç düğümü ve $l = [(m-1)n + 1]/m$ yaprağı vardır.
    - ❑ i iç düğümlü ağacın $n = m \cdot i + 1$ düğümü ve $l = (m-1)i + 1$ yaprağı vardır.
    - ❑ l yapraklı ağacın, $n = (m \cdot l-1)/(m-1)$ düğümü ve $i = (l-1)/(m-1)$ iç düğümü vardır.
  - ▪ Yüksekliği h olan m-ilişkili bir ağaçta en çok m^h yaprak vardır.

- Eğer yüksekliği h olan bir m-ilişikili ağacın l yaprağı var ise, $h \geq \log_m l$ dir. Eğer m-ilişikili ağaç tam ve dengeli ise, $h = \log_m l$ 'dir.

Ağaçların Uygulamaları

- ***İkili Arama Ağacı:*** Bir sıralı köklü ikili ağaçta herbir düğüme; sol alt ağacındaki düğümlerdeki anahtarlardan büyük ve sağ alt ağacındaki düğümlerde bulunan anahtarlardan küçük bir anahtar atanır.(*İkili Arama Ağacı Algoritması.)*
- ***Karar Ağacı*:** Herbir iç düğümün bir karara karşılık geldiği bir köklü ağaçta, kararın herbir olası sonucu için bu düğümlerde bir alt ağaç bulunur.(Örnek, Sahte jetonların bulunması)
- ***Önek Kodları*:** Farklı uzunluktaki bit dizilerini kodlamaya dayalı kodlar, bir harf için bit dizisinin diğerbir harfin ön ekinde olmaması özelliği ile harfleri kodlamakta kullanılır.

Huffman Kodlama Algoritması

Bir ikili ağacı verilen $w_1 \leq w_2 \leq \ldots \leq w_n$ ağırlıklar ile aşağıdaki şekilde yinelemeli olarak oluştur:

1. En küçük iki ağırlığında köklü alt ağaçlı şekilde bir ağaç oluştur. Onların birleştirilmiş ağırlıkları, diğer dalların oluşturulması için ağırlıkların kullanılabileceği bu alt ağacın kökünün ağırlığı olur.
2. Bütün ağırlıklar birleştirilene kadar adım 1'i tekrarla.
3. Herbir iç düğümün 2 dalı 0 ve 1 olarak etiketlenir. Herbir harf, ikili ağaçtan elde edildiği şekilde etiketlerin yolunu alır.

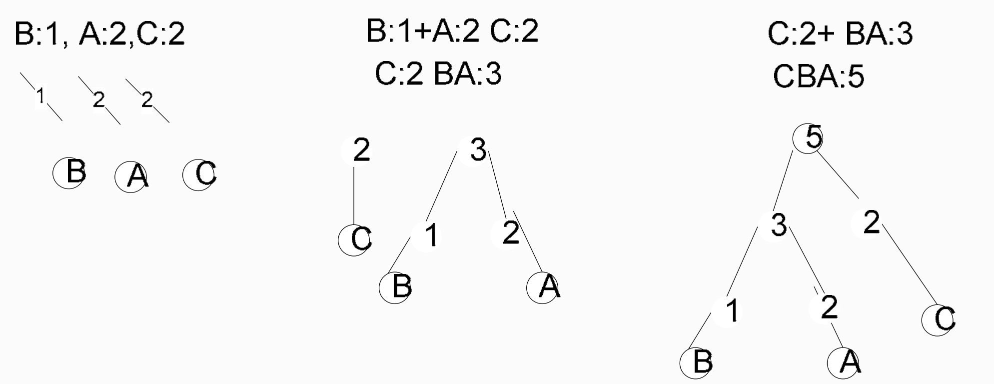
Örnek: Ağacın oluşturulması

İlk önce karakterlerin frekansları (kullanım sıklıkları) hesaplanmalıdır.

Örneğin, elimizdeki veri "**BAACC**" olsun,
B: 1
A: 2
C: 2

En küçük iki frekans toplanır ve frekans tablosu yeniden düzenlenir,

Tek bir ağaç oluşturulana kadar sürekli en küçük frekanslar toplanır,



Şekil 7.15. .Huffman kodlama algoritması örneği

Ağaçların graf teorisinde önemli olmasının bir nedeni tüm bağlı grafların bir ağaç içermesindendir. Buna spanning tree denir ve bütün düğümleri bağlar.

Tanım: G, düğüm kümesi V olan bir bağlı graf olsun. G ‘deki bir spanning tree yine ağaç olan bir alt graftır ve düğüm kümesi V ‘dir.

Teorem 7.10: Bütün bağlı graflar bir spanning tree içerir.

İspat: G bağlı bir graf olmak üzere ; G bir devre içermiyorsa G’ nın kendisi bir spanning tree olduğundan kanıtlanacak bir şey yoktur.

G, bir devre içeriyor diyelim. Devreden bir ayrıt çıkarırsak elimizde hala bağlı bir graf kalır. Eğer yeni graf bir devre içeriyorsa devreden tekrar bir ayrıt çıkarırız. Bu işlemi sonuç grafı T bir devre içermeyinceye kadar devam ettiririz. Hiçbir düğümü kaldırmadığımıza göre T, G ile aynı düğüm kümesine sahip olacaktır ve yukarıdaki işlemin her aşamasında bağlı bir graf elde ederiz. Bu nedenle, T ‘nin kendisi bağlıdır; G için bir spanning tree’ dir.

Düzlem Graflar

Tanım: Düğümleri düzlemde noktalar ve ayrıtları sadece grafın düğümlerinde kesişen doğrular veya yaylar olan grafa düzlem grafı denir.

Bir graf, eğer bir düzlem grafıyla izomorfik ise örneğin düzlemde hiçbir ayrıt’ı kesişmeden bir diyagram ile temsil edilebiliyorsa düzlemsel (planar) graftır.

Euler’ in Formülü

G bağlı düzlemsel bir graf olsun. Düzlemde çizilen G ‘nın diyagramı ‘yüz’ (face) adını verdiğimiz bölgelere ayırır.

Bağlı düzlemsel bir grafın düğümlerinin, ayrıtlarının ve yüzlerinin sayısı arasında bir ilişki kurmak için basit bir formül vardır. Aşağıdaki tablo bu formülü görmek için faydalı olabilir.

Graf	*Düğüm Sayısı*	*Ayrıt Sayısı*	*Yüz Sayısı*
Şekil 7.2 (a)	7	7	2
Şekil 7.3 (a)	9	14	7
Şekil 7.4 (b)	4	7	5
Şekil 7.11 (b)	4	9	7
Herhangi ağaç	n	n-1	1

Bütün bu graflar bağlıdır ve düzlemseldir ve |F|, |E|, |V| sırasıyla yüzlerin, ayrıtların ve düğümlerin sayısı olmak üzere

$$|F| = |E| - |V| + 2$$

ilişkisini sağlarlar. Bu ilişki tüm bağlı düzlemsel graflar için sağlanır ve Euler’ in formülü olarak bilinir.

Teorem 7.11: G, |V| düğümlü, |E| ayrıtlı ve düzlemi |F| yüze veya bölgeye ayıran herhangi bir bağlı düzlemsel graf olsun. Bu durumda, $|F|=|E| -|V| + 2$ olur.

İspat: G’ nın ayrıt sayısına tümevarım yöntemi uygulayarak ispat yapılabilir. $|E|=0$ ise $|V|=1$ (G bağlıdır o halde iki veya daha fazla düğüm olamaz) ve tek bir yüz vardır yani $|F|=1$. Bu nedenle bu durum için teorem doğrudur.

Şimdi, teoremin n ayrıtdan az graflar için de sağlandığını düşünelim. G, n ayrıtlı bağlı düzlemsel graf olsun; yani |E|=n. G bir ağaç ise |V|=n+1 (teorem 7.10) ve |F|=1 o halde, teorem bu durumda da sağlanır. Eğer G bir ağaç değilse G‘deki herhangi bir devreyi seç ve bir ayrıt'ını sil. Sonuçtaki graf G' bağlıdır, düzlemseldir, n-1 ayrıt' i, |V| düğümü, ve |F|-1 yüzü vardır. Tümevarımsal hipoteze dayanarak Euler' in formülü G' için sağlanır.

$$|F|-1 = (|E|-1) - |V| + 2 \quad \text{o halde,}$$
$$|F| = |E| - |V| + 2.$$

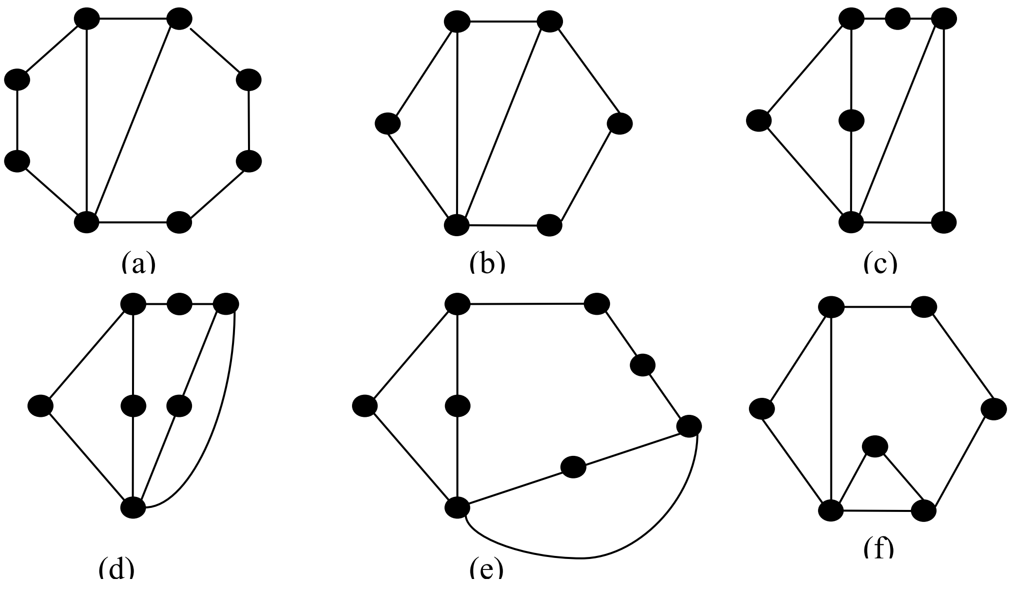
Teorem 7.12: n düğümlü(n ≥3) basit düzlemsel bağlı bir grafta, en fazla (3n-6) ayrıt vardır.

İspat : Eğer n=3 ise ayrıt sayısı en fazla 3'tür. n 3'e eşit veya daha büyük olsun. Düzlemsel grafı $F_1,F_2,\ldots,F_p$ olarak çizelim. F_i ile tanımlanan yüzün ayrıt sayısı r_i olsun. Her bir i için r_i en az üç olur. Böylece $3p \leq (r_1 + r_2 + \ldots + r_p)$ dir. Şimdi, sınırlardaki ayrıtları sayarsak her bir ayrıt ençok iki kere sayılır. Böylece eşitsizliğin sağ tarafı en fazla 2m olur. Burada m garaftaki ayrıt sayısıdır. Böylece, 3p en çok 2m dir. Fakat teorem 7.11'den p(|F|)= 2-n(|V|) +m(|E|) dir. Bu teorem bazı meşhur grafların düzlemsel olmadığını göstermek için kullanılır.

Kuratowski 'nin Teoremi

Tanım: Eğer bir graf, diğer bir grafın ayrıtlarına derecesi 2 olan düğümler ekleyerek veya çıkararak elde edilebiliyorsa bu iki graf homomorfiktir (izomorfik kopyasıdır).

Örnek 7.4: Şekil 7.16 'da gösterilen grafların hepsi homomorfiktir. (a)' daki graftan (b)' dekini elde etmek için 2 düğüm sileriz ve (b)' dekinden (c)' deki grafı elde etmek için bir düğüm sileriz ve iki düğüm ekleriz. (d)' den (e)' yi elde etmek için bir düğüm ekleriz. (e) ve (f)' deki graflar izomorfiktir- herhangi bir düğümün eklenmesine veya çıkarılmasına gerek yoktur.



Şekil 7.16

Teorem 7.13: Bir düzlemsel grafın kromatik sayısı dördü geçemez.

Alıştırmalar

1- Düğüm kümesi {1,2,3,4,5} ve ayrıt kümesi E={{1,2},{1,3},{1,5},{2,3},{3,4},{3,5}, {4,5}} olan grafı çiziniz. Bu grafın komşuluk matrisini bulunuz.

2- Γ ve Σ graflarının düğüm kümeleri ve komşuluk matrisleri sırasıyla şöyledir: $V_\Gamma = \{v_1, v_2, v_3, v_4, v_5\}$ ve $V_\Sigma = \{v_1, v_2, v_4, v_5\}$

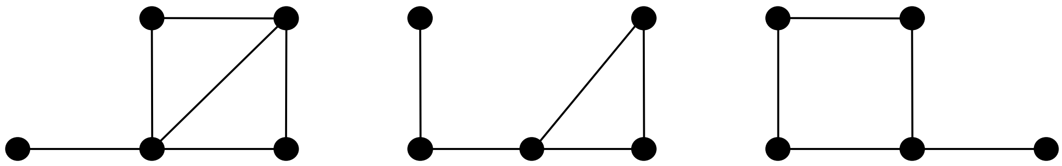
ve

Γ ve Σ graflarını çiziniz ve bu iki graf arasında nasıl bir ilişki vardır açıklayınız.

3- a) Hamilton grafı olan fakat Euler grafı olmayan dört düğümlü bağlı bir graf çiziniz.

b) Ne Hamilton ne de Euler grafı olmayan dört vertexli bağlı bir graf çiziniz.

4- Aşağıdaki 3 şekilden herhangi ikisinin izomorfik olmadığını ve hangi iki tanesinin homomorfik olduğunu gösteriniz.



(a) (b) (c)