CHAPTER 8

GRAFLAR VE ÇOKLU GRAFLAR

Graflar G harfi ile gösterilir ve bir graf köşeler (V) ve kenarlar (E) olmak üzere iki bileşenden oluşur.

i) V bir küme olmak üzere; V = V (G) biçimindeki gösterim, bir G grafındaki köşeleri (vertices), noktaları (points) ya da düğümleri (nodes) ihtiva eder/kapsar.

ii) E bir küme olmak üzere; E = E(G) biçimindeki gösterim, G grafının kenar (edge) olarak adlandırılan ve farklı noktaların sırasız çiftlerini oluşturan bir kümedir.

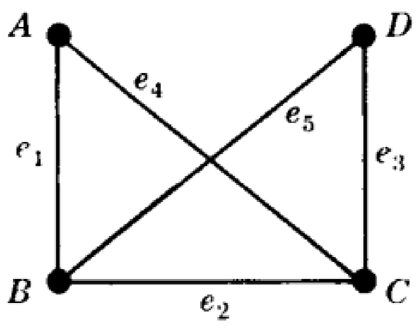
Böylece köşelerden ve kenarlardan oluşan bir grafı temsili olarak G(V , E) biçiminde gösterilir.

ÖRNEK:

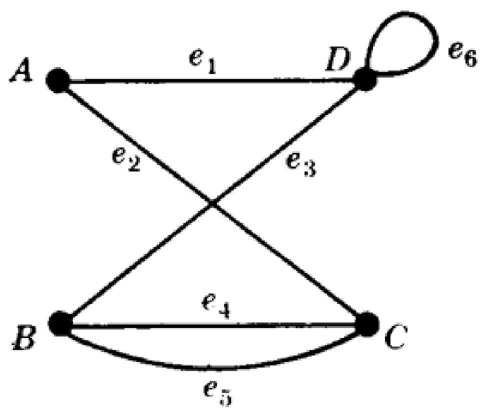
Aşağıda verilen a) gafında G(V , E) notasyonundaki V ve E aşağıdaki gibi ifade edilir.

i) V kümesi A, B, C, D köşelerinden oluşmaktadır.

ii) E kümesi e1 = {A, B}, e2 = {B,C}, e3 = {C, D}, e4 = {A, C}, e5 = {B,D} kenarlarından oluşmaktadır.



(a) Graph



(b) Multigraph

ÇOKLU GRAFLAR

Yukarıda verilen graflardan (b) de gözüken grafı göz önüne alınız. Bu graftaki e4 and e5 kenarlara çoklu kenar denir çünkü aynı köşeleri (endpoit) birleştirirler. Ayrıca e6 kenarına da loop denir, çünkü onun başlangıç ve bitiş köşeleri aynıdır. Bu tür yapıdaki graflara çoklu graflar denir.

KÖŞELERİN DERECESİ

Bir G grafındaki v köşesinin derecesi, deg(v) olarak gösterilir, v köşesindeki kenarların sayısı olarak tarif edilir.

TEOREM. Bir G grafındaki köşelerin derecelerinin toplamı, o grafın kenar sayısının iki katına eşittir.

ÖRNEK:

Yukarıda verilen (a) grafını göz önüne alalım. Bu graftaki her bir köşenin (düğümün) derecelerini hesaplayalım.

deg(A) = 2, deg(B) = 3, deg(C) = 3, deg(D) = 2.

(a) grafının düğümlerinin dereceleri toplamı 10 dur ve (a) grafının kenar sayısı ise (e1 , e2, e3, e4, e5)= 5 tir. Görüldüğü gibi bir graftaki düğümlerin derecelerinin toplamı kenar sayısının iki katı kadardır.

Çoklu grafta loop'u oluşturan kenar e6 iki kez sayılır. Bundan dolayı D düğümünün derecesi deg(D) = 4 çünkü e6 aynı düğümden çıkıp, aynı düğüme geri dönmektedir.

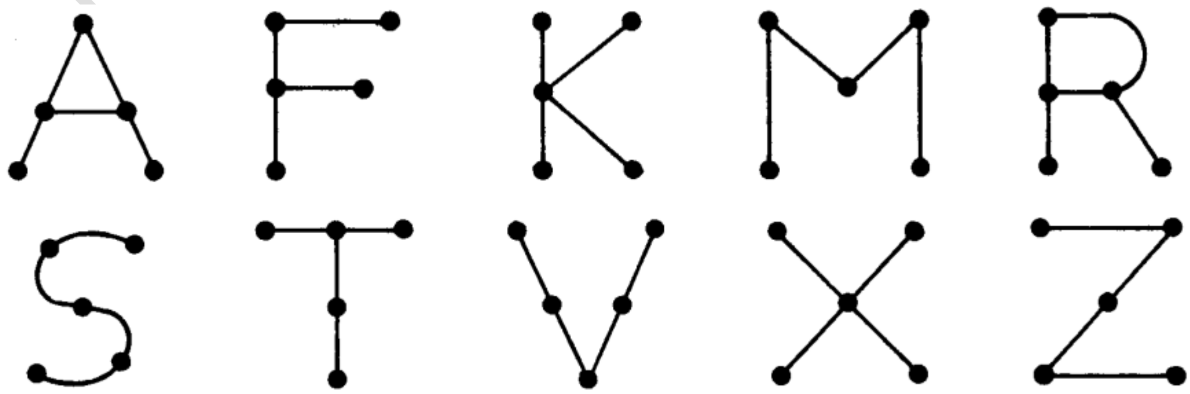
ALT GRAFLAR,

Bir G grafını G = G(V , E) göz önüne alalım. Herhangi bir H = H (V , E) grafı G grafının alt graf olarak adlandırılır eğer H nin sahip olduğu köşeleri ve kenarları G grafında bulunuyorsa. Yani V ⊆ V and E ⊆ E ise.

IZOMORFIK GRAFLAR

Bir G rafı ile G(V , E) ve bir grafı G^* (V^*, E^*) izomorfiktir denir. Eğer, bire-bir bir uyuşma/benzerlik var ise $f : V \to V^*$ öyle ki {u, v} , G nin bir kenarı olup ancak ve ancak eğer {f (u), f (v)} G^* ında bir kenarı olmak üzere. Normal olarak görünümleri farklı olmasına rağmen izomorfik grafları ayırt edemeyiz.

Aşağıdaki şekilde harf olarak verilen on tane graf var. A ve R izomorfik graftır. Ayrıca F ve T izomorfik graftır. K ve X izomorfik graftır. Ayrıca M,S,V ve Z de izomorfik graftır.

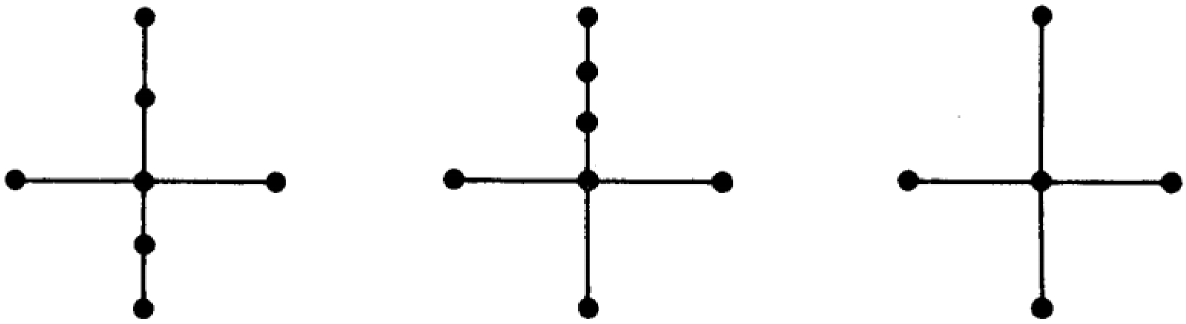


Bu graflara bakarak izomorfik graflar için aşağıdaki açıklamayı yapabiliriz. Bağlantılarına zarar vermeden bir birlerine dönüştürülebilen iki graf izomorfiktir denir.

HOMEOMORFİK GRAFLAR

Verilen herhangi bir G grafında, G grafının herhangi bir kenarına ek düğümler ekleyerek yeni bir graf elde edebiliriz. Bu metotla aynı graftan ya da izomorfik bir graftan G ve G^* grafları elde edilebilirsse bu graflara homeomorfik graf denir.

ÖRNEK: Aşağıdaki graflardan (a) ve (b) grafları izomorfik değil fakat bu iki graf homeomorfiktir. Çünkü bu iki grafı (c) grafından kenarlara düğümler/noktalar ekleyerek elde edebiliriz.



(a) (b) (c)

YOL

Çoklu bir G grafında bir yol köşelerin ve kenarların alternatif bir sırasından oluşur.

$v_0, \quad e_1, \quad v_1, \quad e_2, \quad v_2, \ldots, \; e_{n-1}, \; v_{n-1}, \quad e_n, \quad v_n$

burada her e_i kenarı vi−1 ve vi köşelerini ihtiva eder. Kenarlar üzerindeki n sayısı yolun uzunluğunu oluşturur.

Basit yol: Tüm köşelerin farklı olduğu yola basit yol denir.

Tüm kenarların farklı olduğu bir yola ise trail (iz) denir.

Çevrim (Cycle) başlangıç ve bitiş düğümleri dışındaki tüm düğümleri farklı olan kapalı bir yola çevrim denir.

ÖRNEK: Aşağıdaki şekil (a) da verilen grafa göre oluşturulan sıraları göz önüne alalım:

$$\alpha = (P_4, P_1, P_2, P_5, P_1, P_2, P_3, P_6), \quad \beta = (P_4, P_1, P_5, P_2, P_6),$$
$$\gamma = (P_4, P_1, P_5, P_2, P_3, P_5, P_6), \quad \delta = (P_4, P_1, P_5, P_3, P_6).$$

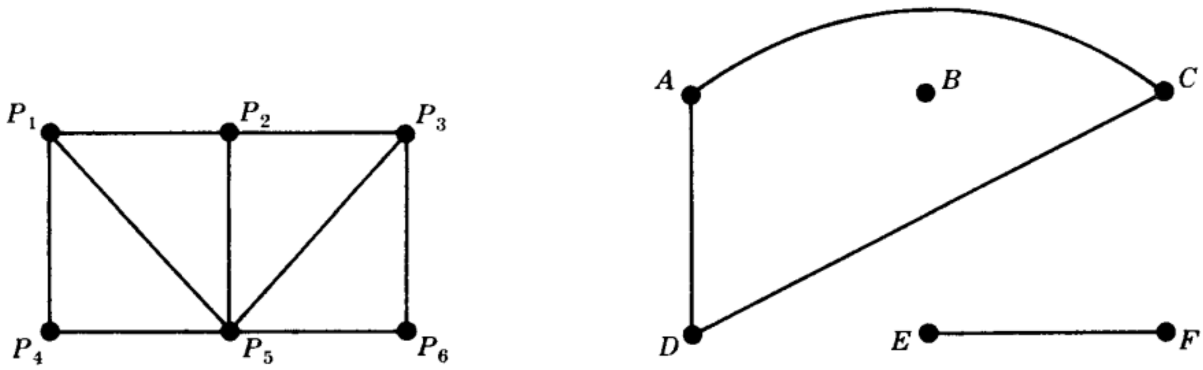
α sırası, P4 dan P6 ya bir yoldur, fakat bir iz (trail) değildir çünkü {P1, P2} iki kez kullanılmıştır.

β sırası, bir yol değildir çünkü {P2, P6} gibi bir kenar yoktur.

γ sırası bir yol (trail) çünkü iki kez kullanılan bir kenara sahip değildir, fakat basit bir yol değildir çünkü P_5 köşesi iki kez kullanılmıştır.

δ sırası P4 dan P6 ya basit bir yoldur, fakat uzunluğu göz önüne alındığında P4 dan P6 en kısa yol değildir.

P4 dan P6 en kısa yol (P4, P5, P6) dır ve uzunluğu 2 olan basit yoldur



(a) (b)

BAĞLANTI (CONNECTIVITY)

Bir G grafının herhangi iki düğümü arasında bir yol var ise bu graf bağlı graftır denir. Yukarıdaki şekilde verilen (a) grafı bağşlı bir graftır fakat (b) grafı bağlı bir graf değildir, çünkü örneğin D düğümü ile E düğümü arasında bir yol yoktur.

BAĞLI ELEMANLAR (CONNECTED COMPONENTS)

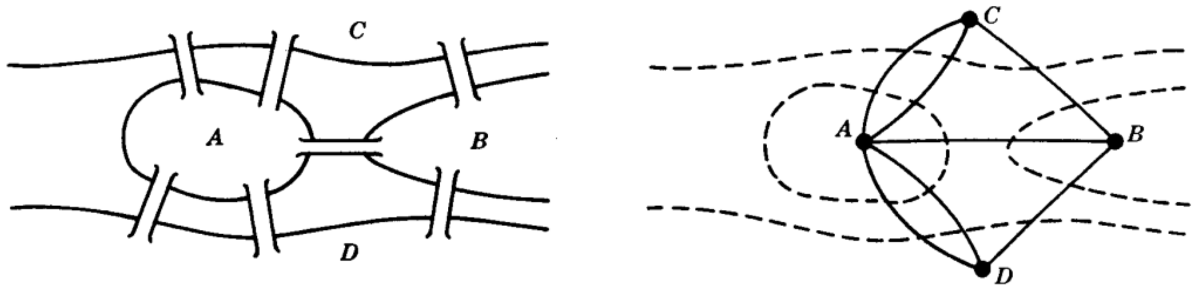
G nin bir graf olduğunu farz edelim. G nin bağlı bir alt grafı olan H ye, G’in bağlı elemanı (connected component) denir Eğer H, G nin bağlı alt graflarından daha büyük (geniş) değilse. Bir g grafı, kendi bağlı elemanlarına bölünebilir.

ÖRNEK: Yukarıdaki (b) grafı [A,C,D], [E,F] ve [B] alt grafları olmak üzere üç tane bağlı elemana sahiptir.

(b) grafındaki B düğümüne izole edilmiş (isolated vertex) denir, çünkü B herhangi bir kenara sahip değildir, başka bir deyişle B düğümünün derecesi deg(B)=0 ’dır.

DOLAŞILABİLİR VE EULERIAN GRAF, KÖNIGSBERG KÖPRÜSÜ

18. yüz yılda doğu Prusya’nın Königsberg kasabasında iki ada ve yedi köprüden oluşan bir ortam vardı. Burada soru: herhangi bir yerden başlayıp herhangi bir yerde bitirmek, herhangi bir kimse köprülerden bir sefer geçmek şartıyla, (bir köprüden iki kez geçmeğe izin verilmiyor) kasabanın karşısına geçmek mümkün müdür? Königsberg insanları bu problemi zamanın ünlü İsviçreli matematikçisi L.Euler’e yazarak çözümü olup olmadığını sorarlar. Euler 1736’da böyle bir dolaşımın mümkün olmadığını ispatlar. Euler bu problemi adaları birer nokta (düğüm) ve köprüleri de birer kenar olarak modelleyerek aşağıdaki gibi bir grafa dönüştürmüştür.



(*a*) **Königsberg in 1736** (*b*) **Euler’s graphical representation**

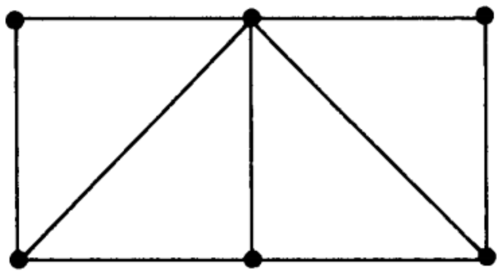
Görüldüğü gibi (b) grafı bir çoklu graftır. Eğer “çizgiler kesilmeksizin ve kenarlar tekrar edilmeksizin” yani tüm düğümleri içeren ve her kenarı sadece bir kez kullanan bir yolu olan çoklu graf “traversable” dır yani “dolaşılabilir” dir denir. Böyle bir yol bir iz (trail) olmalı çünkü iki kez kullanılan kenar yok ve dolaşılabilir iz olarak adlandırılır. Açıkçası, dolaşılabilir bir çoklu graf sınırlı ve bağlı (connected) olmalıdır.

Şimdi de Euler’ın çoklu grafın dolaşılabilir olmadığını ve böylece Königsberg köprüsünde yürümenin mümkün olmadığını nasıl ispatladığını göstereceğiz. Bir düğümün derecesine göre çift ya da tek olabildiğini hatırlayalım. Sonuç olarak Euler iki tek dereceli düğümden fazla düğümü olan bir çoklu graf dolaşılamaz olduğunu gösterdi. Konigsberg köprüsü problemi dört tek dereceli düğüme sahiptir.

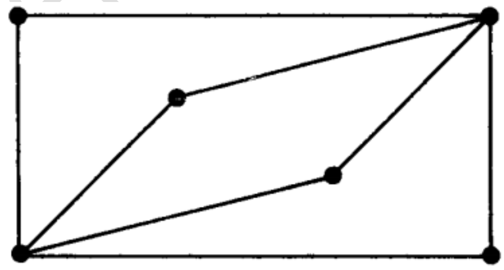
Bu sonuç Euler teoremi olarak bilinir. Bir G grafı, eğer kapalı dolaşılabilir bir trail (yol) var ise/sahipse bu yola Euler yolu denir. Başka bir ifade ile sınırlı bağlı bir graf ancak her düğüm çift dereceye sahipse Eulerian'dır.

HAMİLTON GRAF

Yukarıda Euler için tartışılan/ifade edilenler dolaşılır kenarları vurgulamaktadır, Hamilton'da ise ziyaret edilen düğümlere odaklanılır. Bir G grafındaki Hamilton devresi, kapalı bir yoldur öyle ki bu yol G grafında bulunan düğümleri sadece bir kez dolaşır/ziyaret eder. Böyle bir kapalı yol döngü olmalıdır. Eğer bir G grafı Hamilton devresine sahipse bu graf Hamilton grafı olarak adlandırılır. Burada şunu hatırlatmak gerekir; Euler devresi her bir kenarı sadece bir kez dolaşır, fakat düğümlerden birden fazla kez geçebilir, ancak Hamilton devresi düğümleri sadece bir kez ziyaret eder, fakat kenarlardan birden fazla kez geçebilir. Aşağıdaki şekilde verilen graflarda Hamilton ve Euler devresine ait örnekler verilmiştir.



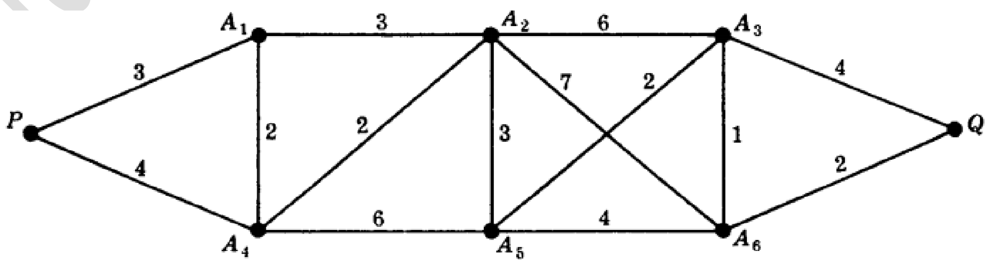
(*a*) Hamiltonian and non-Eulerian



(*b*) Eulerian and non-Hamiltonian

ETİKETLİ VE AĞIRLIKLI GRAFLAR

Bir G grafının kenarlarına veya köşelerine bir veri (data) atanmış ise bu graf etiketli veya ağırlıklı graf olarak adlandırılır. Özellikle bir G grafının her kenarına (e) negatif olmayan bir sayı atanırsa bu sayıya o kenarın ağırlığı w(e) ya da uzunluğu denir. Böyle bir graf örneği aşağıdaki şekilde verilmiştir.



Böyle ağırlıklı bir G grafında bir yolun ağırlığı ya da uzunluğu yolun üzerindeki kenarların ağırlıklarının toplamıdır. Graf teorisinde önemli problemlerden birisi de en kısa yol problemidir. Yani verilen herhangi iki köşe arasındaki minimum ağırlıklı (uzunluklu) yolu bulmaktır.

Yukarıdaki şekilde P ve Q arasındaki böyle bir yol en kısa uzunluğu şu şekilde yazılır:

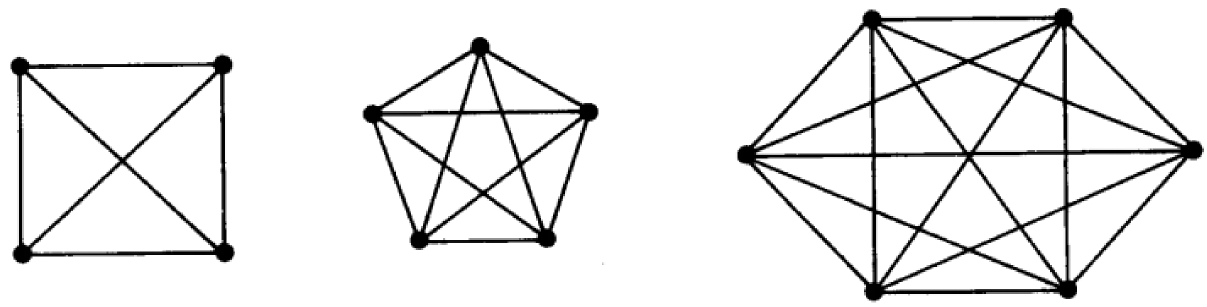
(P , A1, A2, A5, A3, A6, Q)

KOMPLE, DUZENLİ VE IKI PARÇA (BIPARTITE) GRAFLAR

Bir çok farklı graf olmasına rağmen bu bölümde sadece komple, düzenli ve iki parça graflar göz önüne alınmıştır.

KOMPLE:

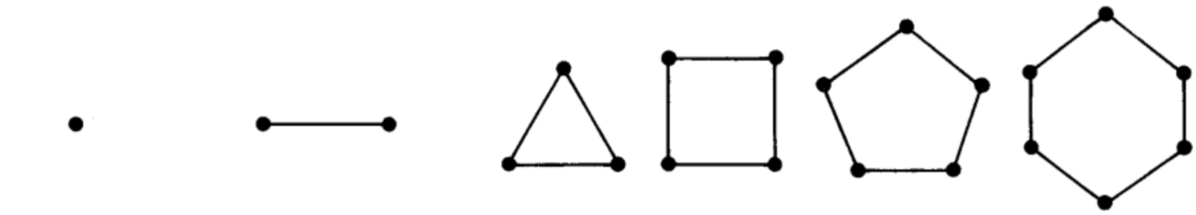
Eğer bir G grafının tüm düğümleri yine aynı G grafının tüm düğümleri ile bağlı ise bu G grafına komple graf denir. Böylece komple bir G grafı bağlı (connected) graftır. n köşeli bir komple graf K_n ile temsil edilir. Aşağıdaki şekilde K_1 köşeliden K_6 köşeliye kadar olan graflar gözüküyor.



K_4 K_5 K_6

DÜZENLİ GRAFLAR

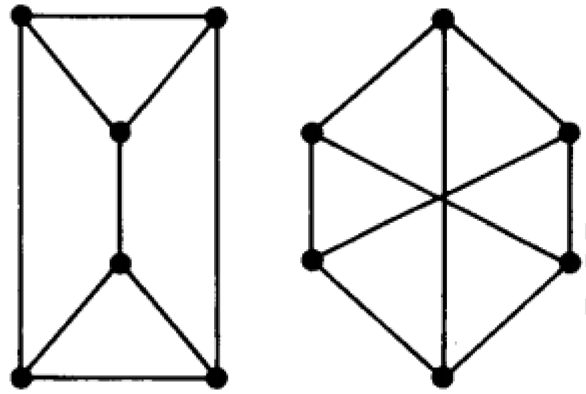
Bir G grafının her düğümünün derecesi aynı ise bu durumda G grafı düzenli bir graftır denir. Düzenli graflar k-regular olarak gösterilir. Yani derece değeri k olan düzenli graf. Aşağıdaki şekilde düzenli graflar yer almaktadır.



(i) 0-regular (ii) 1-regular (iii) 2-regular

Burada 0,1 ve 2 dereceli bağlı düzenli graflar yer almaktadır. 0-düzenli graf bir düğümlü ve kenarsız bir graf olup önemsiz (trivial) graf olarak ta adlandırılır. 1-düzenli graf ise iki düğümlü ve bir kenardan oluşan bir graftır. Bağlı 2-düzenli graf ise n düğümlü bir graf olup, n düğümlü bir döngüden oluşur.

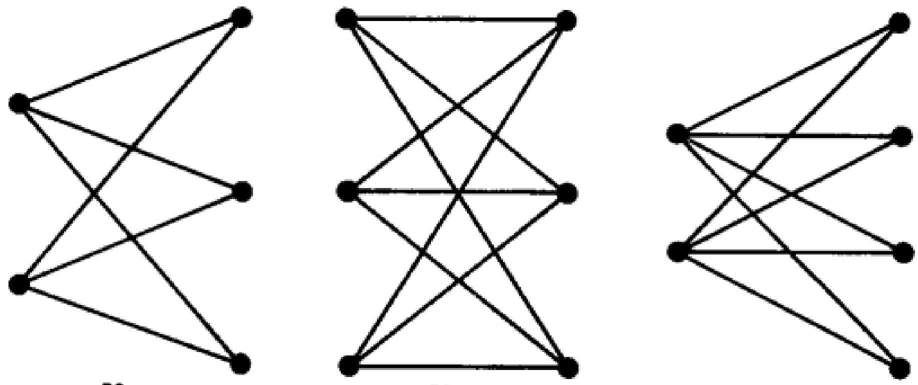
3-dereceli düzenli grafların düğüm sayısı çift olmalıdır, çünkü daha önce verilen teoreme göre bir G garafının düğümlerinin dereceleri toplamı kenar sayısının iki katına eşit olduğunu biliyoruz. Örneğin aşağıda verilen şekilde iki tane bağlı 3-regular graf altı köşelidir.



3-regular

BİPARTITE GRAFLAR

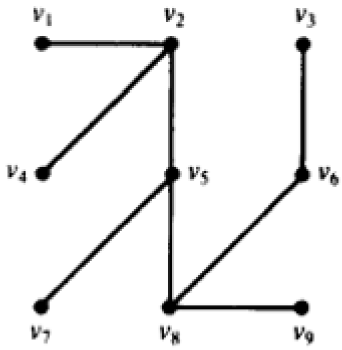
Bir G grafının düğümleri (V) eğer M ve N gibi alt düğümlere bölüne bilirse öyle ki G grafının her kenarı M düğümünü N düğümüne bağlar, böyle graflara iki parçalı graflar denir. Komple iki parçalı graf ile , her M düğümü, her N düğümü ile bağlı olan bir graf kastedilmektedir ve bole bir graf Km,n notasyonu ile gösterilir. Burada m, M deki dğümlerin sayısını, n de N deki düğümlerin sayını ifade etmektedir. Standar olarak m ≤ n kabul edilir. Aşağıdaki şekilde K2,3, K3,3, and K2,4 grafları gösterilmektedir.



$K_{2,3}$ $K_{3,3}$ $K_{2,4}$

AĞAÇ GRAFLAR

Eğer T bir bağlı ve döngüleri olmayan bir graf ise bu grafa ağaç denir. Aşağıdaki şekilde verilen G grafı döngüleri olmayan bir graftır, böylece G grafının elemanları (komponentleri) bir ağaç oluşturur. Tek düğümlü ve kenarı olmayan ağaca degenerate ağaç denir.



(a)

Bir T ağacını göz önüne alalım: T grafında iki düğüm arasında sadece bir yol vardır, bunun aksi durumunda (yani iki yol olması halinde) bir döngü oluşur.

a) T grafında (u,v) arasında bir kenar yok olduğunu farz edelim ve e={u,v} kenarını T'ye eklediğimizde, bu kenar u'dan v'ye basit bir yol ve bir döngü oluşturur. Böylece T grafı artık bir ağaç olmaz.

b) Diğer yandan, T'de e={u,v} yolunun var olduğunu kabul edelim ve T'den e yi silelim. T bağlı olmadığından (connected) çünkü u'dan v'ye bir yol yok, bu durumda T artık bir ağaç değildir.

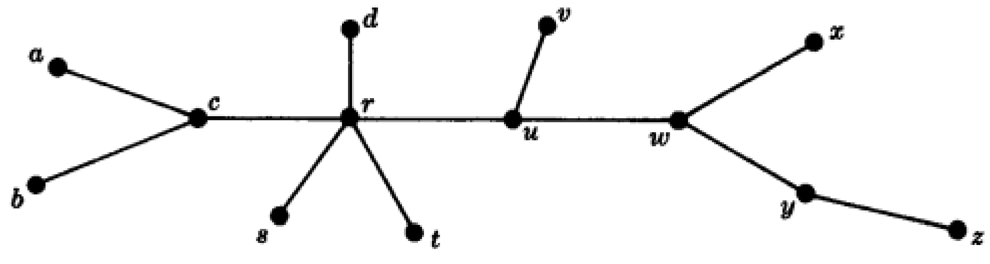
Teorem: n>1 olacak şekilde düğüm sayısı olan bir G grafını göz önüne alalım. Bu durumda aşağıdaki eş değerlikler yazılabilir:

i) G bir ağaçtır

ii) G döngü içermiyor ve n-1 kenara sahiptir.

iii) G bir bağlı graftır ve n-1 kenara sahiptir.

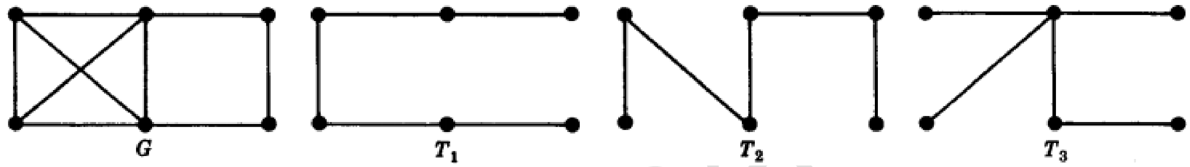
Bu teorem bize n düğümlü sınırlı bir T ağacının n-1 kenara sahip olması gerektiğini söyler. Örneğin; Aşağıdaki şekilde (a) da 9 düğüm ve 8 kenar var ve (b) de ise 13 düğüm ve 12 kenar var.



(b)

SPANNING TREES (AĞAÇLAR)

Bağlı bir G grafının alt grafı olan T'ye G grafının spanning tree'si denir eğer T bir ağaç ve G grafının tüm düğümlerini içeriyorsa. Aşağıdaki şekilde bağlı bir G grafı ve G grafının T1,T2 ve T3 olmak üzere spanning tree'leri gözükmektedir.



Minimum Spanning Tree

G nin ağırlıklı bağlı bir graf olduğunu farz edelim. Yani, G'nin her kenarına negatif olmayan ve ağırlık olarak adlandırılan bir sayı atanmıştır. Sonra, G'nin herhangi bir T spannin tree'si (ağacı), T'deki kenarların ağrılıklarının toplanmasıyla elde edilen toplam bir ağırlığa atanır. G grafının minimum spanning tree'si toplam ağrılığı mümkün olduğunca az olabilen bir spanning tree dir. Aşağıdaki algoritmalar ağırlıklı ve bağlı bir G grafında minimum spanning T ağacını bulmamıza imkan verir.

Algortima-1: Input: n düğümlü ağırlıklı bağlı bir G grafı

Adım 1. G2nin kenar ağırlıklarını azalacak şekilde sıraya diz

Adım 2. n-1 kenar bozulmayana kadar sıralanmış kenarları ardarda silme işlemini yap

Adım 3. Çık.

Algoritma-2 (Kruskal): Input: n düğümlü ağırlıklı bağlı bir G grafı

Adım 1. G2nin kenar ağırlıklarını artacak şekilde sıraya diz

Adım 2. Sadece G'nin düğümlerinden başlayarak ve işlemi ardarda yaparak, döngü oluşmayana ve n-1 kenar sayısı kuralı korunana kadar kenar ekle.

Adım 3. Çık.

Bir grafta bir ya da daha fazla kenar aynı ağırlığa sahipse, farklı miimal spanning tree'ler oluşabilir. Böyle bir durumda Algoritma-1 ve Algoritma-2'de birinci adımda kenarların düzenlenmesi tek (unique) olmadığından, bundan dolayı aşağıdaki örnekte gösterildiği gibi farklı minimal spanning tree'ler meydana gelebilir.

ÖRNEK: Aşağıda verilen ağırlıklı Q grafının minimal spanning tree'sini bulunuz.

Dikkat edilirse grafın altı düğümü var, bundan dolayı minimal spanning tree beş kenara sahip olacak. Burada ilk olarak birinci algoritmayı uygulayalım.

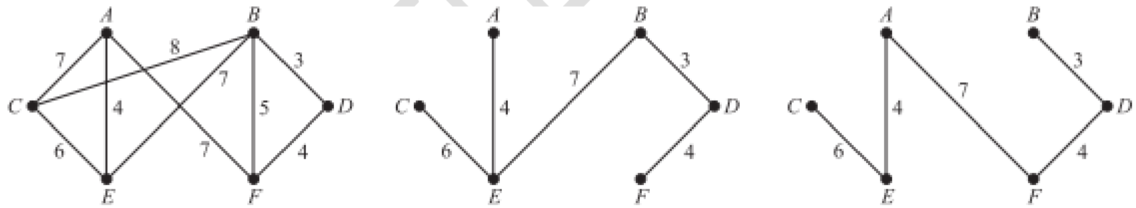
Adım.1-İlk olarak ağırlıklar azalacak şekilde kenar sıralanır ve sonra beş kenar kalana kadar Q'nun bağlantısını koparmadan ardıl olarak kenarları sil. Bu aşağıdaki neticeyi verir.

Edges	BC	AF	AC	BE	CE	BF	AE	DF	BD
Weight	8	7	7	7	6	5	4	4	3
Delete	Yes	Yes	Yes	No	No	Yes			

Böylece, Q'nun minimal spanning tree ağacı, kenarlarıda elde edildi/getirildi.

BE, CE, AE, DF , BD

Spannin tree 24 birim ağırlığa sahiptir ve aşağıdaki şekil (b) gözüküyor.



(a) (b) (c)

İkinic algoritmayı uyguladığımızda

Adım.1. İlk olarak kenar ağrılıkları, artacak şekilde sırala, ve daha sonra döngü oluşmadan ardıl olarak kenarları ekleriz. Bu algoritma aşağıdaki çıktıyı vermiştir.

Edges	BD	AE	DF	BF	CE	AC	AF	BE	BC
Weight	3	4	4	5	6	7	7	7	8
Add?	Yes	Yes	Yes	No	Yes	No	Yes		

Q'nun minimal spanning tree aşağıdaki kenarları ihtiva eder.

Örmek: BD, AE, DF , CE, AF