Ayrık Matematik

Çizgeler

H. Turgut Uyar   Ayşegül Gençata Yayımlı   Emre Harmancı

2001-2013

Lisans

Konular

Çizgeler

Tanım

çizge: $G = (V, E)$

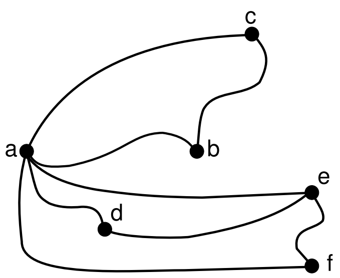
- V: düğüm kümesi
- $E \subseteq V \times V$: ayrıt kümesi

- $e = (v_1, v_2) \in E$ ise:
  - v_1 ve v_2 düğümleri e ayrıtının *uçdüğümleri*
  - e ayrıtı v_1 ve v_2 düğümlerine *çakışık*
  - v_1 ve v_2 düğümleri *bitişik*
- hiçbir ayrıtın çakışmadığı düğüm: *yalıtılmış düğüm*

Çizge Örneği

Düğümler: {1,2,3,4}
Ayrıtlar: (1,2),(2,3)
4. düğüm yalıtılmış

Örnek



$$
\begin{aligned}
V &= \{a, b, c, d, e, f\} \\
E &= \{(a, b), (a, c), \\
&\quad (a, d), (a, e), \\
&\quad (a, f), (b, c), \\
&\quad (d, e), (e, f)\}
\end{aligned}
$$

Yönlü Çizgeler

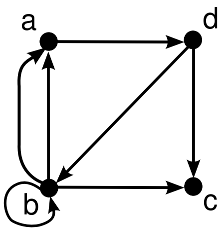
Tanım

yönlü çizge: $D = (V, A)$

- $A \subseteq V \times V$: yay kümesi
- *başlangıç* ve *bitiş* düğümleri

Yönlü Çizge Örneği

Örnek



Çoklu Çizgeler

Tanım

koşut bağlı ayrıtlar: aynı iki düğüm arasındaki ayrıtlar

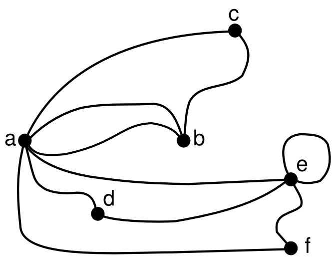
tek-çevre: aynı düğümde başlayan ve sonlanan ayrıt

yalın çizge: koşut bağlı ayrıt ya da tek-çevre içermeyen çizge

çoklu çizge: yalın olmayan çizge

Çoklu Çizge Örneği

Örnek



- koşut bağlı ayrıtlar: (a, b)
- tek-çevre: (e, e)

Altçizge subgraph

Tanım

$G' = (V', E')$ çizgesi $G = (V, E)$ çizgesinin bir altçizgesi:

- $V' \subseteq V$
- $E' \subseteq E$
- $\forall (v_1, v_2) \in E' \; v_1, v_2 \in V'$

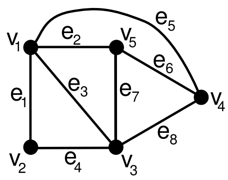
Gösterilim

- *çakışıklık matrisi*:
  - satırlara düğümler, sütunlara ayrıtlar
  - ayrıt düğüme çakışıksa 1, değilse 0
- *bitişiklik matrisi*:
  - satırlara ve sütunlara düğümler
  - hücrelere düğümler arasındaki ayrıt sayısı

Çakışıklık Matrisi Örneği

Incidence Matrix

Örnek

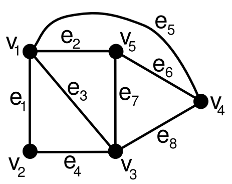


	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
v_1	1	1	1	0	1	0	0	0
v_2	1	0	0	1	0	0	0	0
v_3	0	0	1	1	0	0	1	1
v_4	0	0	0	0	1	1	0	1
v_5	0	1	0	0	0	1	1	0

Bitişiklik Matrisi Örneği

adjacency matrix

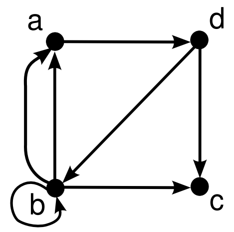
Örnek



	v_1	v_2	v_3	v_4	v_5
v_1	0	1	1	1	1
v_2	1	0	1	0	0
v_3	1	1	0	1	1
v_4	1	0	1	0	1
v_5	1	0	1	1	0

Bitişiklik Matrisi Örneği

Örnek



g(a)=2, c(a)=1
g(b)=2, c(b)=4
g(c)=2, c(c)=0
g(d)=1, c(d)=2
E=7

to

from

	a	b	c	d
a	0	0	0	1
b	2	1	1	0
c	0	0	0	0
d	0	1	1	0

Kerte

derece

Tanım

kerte: düğüme çakışan ayrıtların sayısı

Teorem

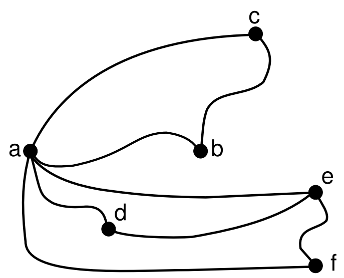
v_i düğümünün kertesi d_i olsun

$$|E| = \frac{\sum_i d_i}{2}$$

Kerte Örneği

çizgenin derece değeri

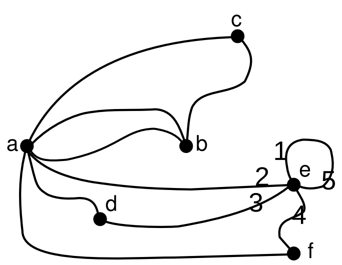
Örnek (yalın çizge)



$$\begin{aligned} d_a &= 5 \\ d_b &= 2 \\ d_c &= 2 \\ d_d &= 2 \\ d_e &= 3 \\ d_f &= 2 \\ Toplam &= 16 \\ |E| &= 8 \end{aligned}$$

Kerte Örneği

Örnek (çoklu çizge)



$$\begin{aligned} d_a &= 6 \\ d_b &= 3 \\ d_c &= 2 \\ d_d &= 2 \\ d_e &= 5 \\ d_f &= 2 \\ Toplam &= 20 \\ |E| &= 10 \end{aligned}$$

Yönlü Çizgelerde Kerte

- kerte ikiye ayrılır
  - *giriş kertesi*: $d_v{}^i$
  - *çıkış kertesi*: $d_v{}^o$
- giriş kertesi 0 olan düğüm: *kaynak* source
- çıkış kertesi 0 olan düğüm: *kuyu* sink
- $\sum_{v \in V} d_v{}^i = \sum_{v \in V} d_v{}^o = |A|$

Ç (tüm düğümlerin kerte toplamı)= Ç (çift olanlar) + Ç (tek olanların kerte sayisi da cift)
2E

Kerte

Teorem
Yönsüz bir çizgede kertesi tek olan düğümlerin sayısı çifttir.

Tanıt.

- t_i: kertesi i olan düğümlerin sayısı
  $2|E| = \sum_i d_i = 1t_1 + 2t_2 + 3t_3 + 4t_4 + 5t_5 + \dots$
  $2|E| - 2t_2 - 4t_4 - \dots = t_1 + t_3 + \dots + 2t_3 + 4t_5 + \dots$
  $2|E| - 2t_2 - 4t_4 - \dots - 2t_3 - 4t_5 - \dots = t_1 + t_3 + t_5 + \dots$
- sol yan çift olduğuna göre sağ yan da çifttir

$\square$

Düzenli Çizgeler

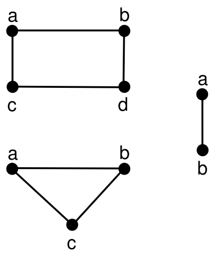
regular graphs

Tanım
düzenli çizge: bütün düğümlerin kertesi aynı

- n-düzenli: bütün düğümlerin kertesi n

Düzenli Çizge Örnekleri

Örnek



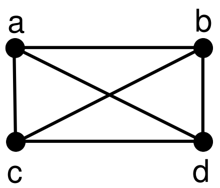
Tam Bağlı Çizgeler

Tanım
$G = (V, E)$ çizgesi tam bağlı:

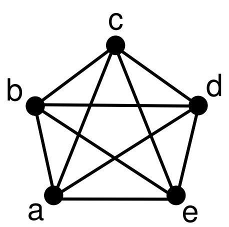
- $\forall v_1, v_2 \in V\ (v_1, v_2) \in E$
- her düğüm çifti arasında ayrıt var
- K_n: n düğümlü tam bağlı çizge

Tam Bağlı Çizge Örnekleri

Örnek (K_4)



Örnek (K_5)



hepsinin n-1 tane kertesi var

İki Parçalı Çizgeler

Tanım
$G = (V, E)$ çizgesi iki parçalı:

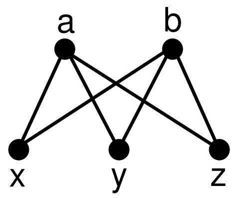
- $\forall (v_1, v_2) \in E\ v_1 \in V_1 \wedge v_2 \in V_2$
- $V_1 \cup V_2 = V,\ V_1 \cap V_2 = \emptyset$
- *tam bağlı iki parçalı*: $\forall v_1 \in V_1\ \forall v_2 \in V_2\ (v_1, v_2) \in E$
- $K_{m,n}$: $|V_1| = m,\ |V_2| = n$

planar: düzlemsel

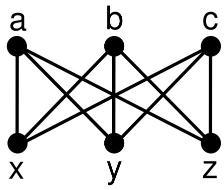
İzomorfizm, iki matematiksel yapı arasında işlemleri ve ilişkileri koruyan bire bir ve örten bir eşleme bulunması durumudur.

Tam Bağlı İki Parçalı Çizge Örnekleri

Örnek ($K_{2,3}$)



Örnek ($K_{3,3}$)



İzomorfizm

Örten: Her y kapsanmalı
Birebir: Her y için sadece 1 x değeri

Tanım
$G = (V, E)$ ile $G^{\star} = (V^{\star}, E^{\star})$ çizgeleri izomorfik:

- $\exists f : V \to V^{\star}$ $(u, v) \in E \Rightarrow (f(u), f(v)) \in E^{\star}$
- f birebir ve örten
- G ile $G^{\star}$ aynı şekilde çizilebilir

planar=düzlemsel

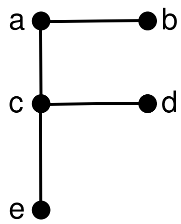
https://en.wikipedia.org/wiki/Graph_isomorphism

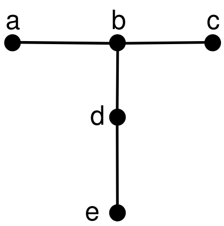
Graph is planar since no link is overlapping with another.
Graph is non-planar since many links are overlapping

İzomorfizm Örneği

tum dugumlerin sayisi ve derecesi esit olmalı

Örnek





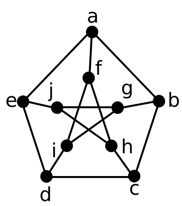
- $f = \{(a, d), (b, e), (c, b), (d, c), (e, a)\}$

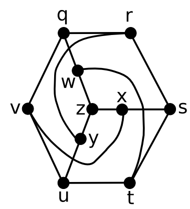
https://www.youtube.com/watch?v=RoDR40UG--s

İzomorfizm Örneği

düğüm sayısı, düğüm sayılarının derece sayıları ve toplam derece değerleri
Bu garanti midir????

Örnek (Petersen çizgesi)





- $f = \{(a, q), (b, v), (c, u), (d, y), (e, r),$
$(f, w), (g, x), (h, t), (i, z), (j, s)\}$

https://www.youtube.com/watch?v=0RzpS1SXJ68

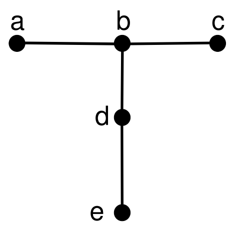
Homeomorfizm

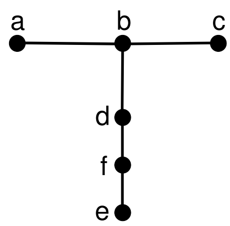
Tanım
$G = (V, E)$ ile $G^{\star} = (V^{\star}, E^{\star})$ çizgeleri homeomorfik:

- $E^{\star}$ kümesindeki ayrıtlardan bazılarının ek düğümlerle bölünmüş olmaları dışında G and $G^{\star}$ çizgeleri izomorfik

Homeomorfizm Örneği

Örnek





Dolaşı walk

Tanım

dolaşı: bir başlangıç düğümünden (v_0) bir varış düğümüne (v_n) bir düğüm ve ayrıt sekansı

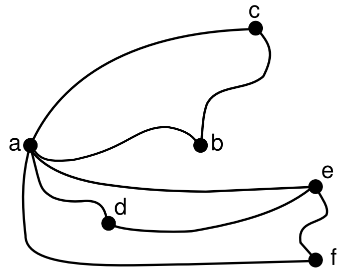
$$v_0, e_1, v_1, e_2, v_2, e_3, v_3, \ldots, e_{n-1}, v_{n-1}, e_n, v_n$$

$e_i = (v_{i-1}, v_i)$

- ayrıtları yazmaya gerek yok
- uzunluk: dolaşıdaki ayrıt sayısı
- $v_0 \neq v_n$ ise açık, $v_0 = v_n$ ise kapalı

Dolaşı Örneği

Örnek



$(c, b), (b, a), (a, d), (d, e), (e, f), (f, a), (a, b)$

c, b, a, d, e, f, a, b

uzunluk: 7

Gezi trial

tekrarsız kenar aynı düğüme dönen

Tanım

gezi: ayrıtların yinelenmediği dolaşı

- devre: kapalı gezi
- kapsayan gezi: çizgedeki bütün ayrıtlardan geçen gezi

Trail (kenar tekrar edilmez yol):

Bir graf üzerinde **aynı kenarı iki kez kullanmadan** gidilen yürüyüştür.

Örnek: A–B–C–D (hiçbir kenar tekrar edilmez).

Circuit (kapalı trail):

Başladığı düğümde biten **kapalı trail**'dir.

Örnek: A–B–C–A (kenarlar tekrar edilmez, başlangıç = bitiş).

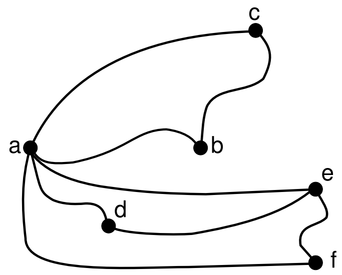
Spanning Trail (tüm kenarları kapsayan trail):

Grafın **bütün kenarlarını en az bir kez** kapsayan trail'dir.

Örnek: Bir grafın her kenarından geçecek şekilde çizilen tek bir yürüyüş.

Gezi Örneği

Örnek



tekrarlı ayrıt yok

$(c, b), (b, a), (a, e), (e, d), (d, a), (a, f)$

c, b, a, e, d, a, f

kapsayan gezi mi?
(a,c) yok x

Yol
path

aynı düğüme dönen tekrarsız node

Tanım

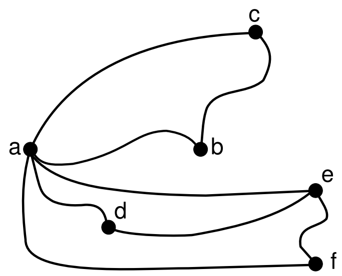
yol: düğümlerin yinelenmediği dolaşı

- çevre: kapalı yol cycle
- kapsayan yol: çizgedeki bütün düğümlere uğrayan yol
  spanning

A-B-C-D-A DÖNGÜ

Yol Örneği

Örnek



$(c, b), (b, a), (a, d), (d, e), (e, f)$

c, b, a, d, e, f

Bağlılık connected graphs

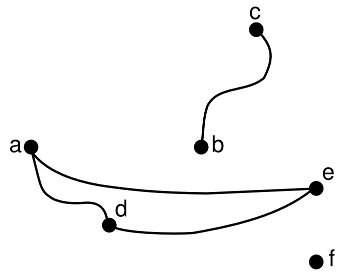
Tanım

bağlı çizge: her düğüm çifti arasında bir yol var

- bağlı olmayan bir çizge bağlı bileşenlere ayrılabilir

Bağlı Bileşen Örneği

Örnek



- çizge bağlı değil:
  a ile c arasında yol yok
- bağlı bileşenler:
  a, d, e
  b, c
  f

Uzaklık distance

Tanım

v_i ile v_j düğümleri arasındaki uzaklık:

- v_i ile v_j arasındaki en kısa yolun uzunluğu

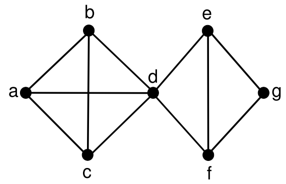
Tanım

çap: çizgedeki en büyük uzaklık diameter

en kısa uzaklıklar içerisindeki
en uzun yol

Uzaklık Örneği

Örnek



- a ile e düğümlerinin uzaklığı: 2
- çap: 3

Kesitleme Noktası cut points

Tanım

$G - v$:

- G çizgesinden v düğümü ve ona çakışık bütün ayrıtların çıkarılmasıyla elde edilen çizge

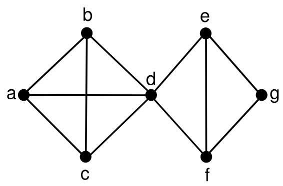
Tanım

v düğümü G çizgesi için bir kesitleme noktası:

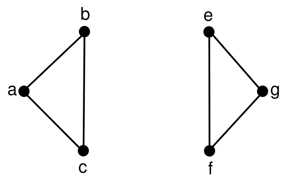
- G bağlı ama $G - v$ bağlı değil

Kesitleme Noktası Örneği

G



$G - d$



bu konu bayes ağlarında çok önemli
ağ oluşumu açısından

Yönlü Dolaşılar

directed walks

- yönsüz çizgelerle aynı
- yayların yönleri gözardı edilirse: *yarı-dolaşı, yarı-gezi, yarı-yol*

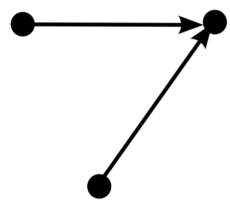
çift taraflı olmayan

Zayıf Bağlı Çizge

geçişlilik aramak

weakly

Örnek



Tanım

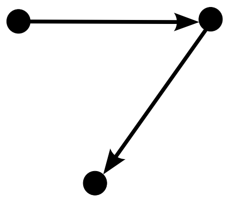
zayıf bağlı:
her düğüm çifti arasında
bir yarı-yol var

kenar yok ama ortak düğüm var

Tek-Yönlü Bağlı Çizge

unilaterally

Örnek



Tanım

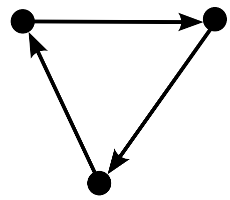
tek-yönlü bağlı:
her düğüm çifti arasında
birinden diğerine yol var

dolaylı yürüyüş var

Güçlü Bağlı Çizge

strongly

Örnek



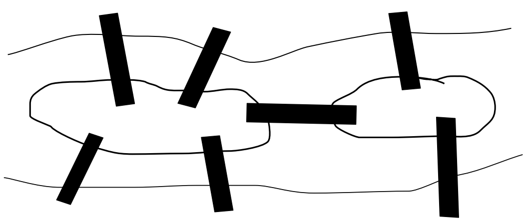
Tanım

güçlü bağlı:
her düğüm çifti arasında
her iki yönde yol var

direkt yollar var (tek yönlü)

Königsberg Köprüleri

https://www.youtube.com/watch?v=tJZk8szUXl4

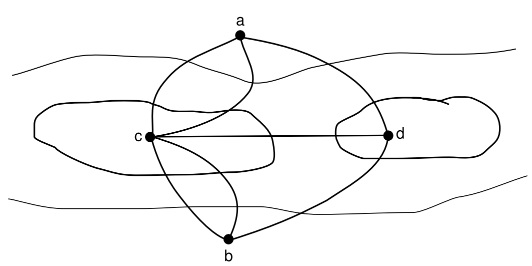


- bütün köprülerden bir kere geçilerek başlangıç noktasına dönülebilir mi?

Geçit Veren Çizge

Tanım

G geçit verir: G üzerinde kapsayan bir gezi düzenlenebilir



1'de aynı

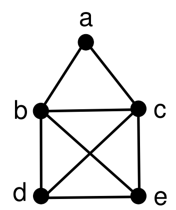
1. durum, başlangıç ve bitiş noktası aynı ise tüm düğüm dereceleri çift
2. durum, başlangıç ve bitiş noktası farklı ise sadece başlangıç ve bitiş düğümleri tek geri kalan noktaların kerteleri çift olmalı

Geçit Veren Çizge

- kertesi tek olan bir düğüm varsa gezinin ya başlangıç düğümü ya da varış düğümü olmalı
- başlangıç düğümü ve varış düğümü dışındaki bütün düğümlerin kerteleri çift olmalı

Geçit Veren Çizge Örneği

Örnek

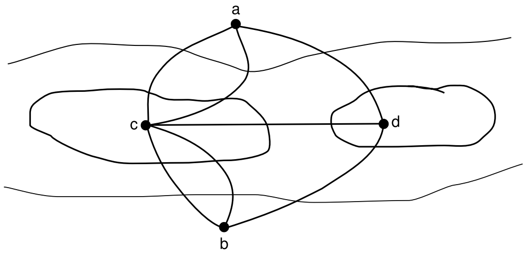


- a, b ve c düğümlerinin kerteleri çift
- d ve e düğümlerinin kerteleri tek
- d düğümünden başlayıp e düğümünde biten (ya da tersi) bir kapsayan gezi oluşturulabilir: $d, b, a, c, e, d, c, b, e$

db-bc-ac-ab-be-ce-cd-de
ec-ca-ab-bc-cd-db-be-ed

tekrarlı ayrık yok
gezi

Königsberg Köprüleri



- bütün düğümlerin kerteleri tek: geçit vermez

gezi: ayrıtlara tekrar ugranmaz

kapsayan gezi: cizgedeki butun ayrıtlardan gecen gezi

Euler Çizgeleri

kapalı:başladığı düğümde biten

Tanım

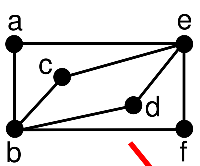
Euler çizgesi: kapalı, kapsayan bir gezi düzenlenebilen çizge

- G bir Euler çizgesi $\Leftrightarrow$ G'deki bütün düğümlerin kerteleri çift

Euler: Başlangıç ve bitiş noktaları:
1. aynı ise: bütün düğümlerin çift sayıda derecesi olmalı
2. farklı ise: başlangıç ve bitiş düğümleri tek sayıda , geri kalan düğümlerin çift sayıda derecesi olmalı

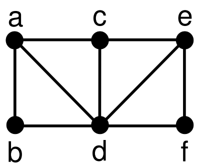
Euler Çizgesi Örnekleri

Örnek (Euler çizgesi)



ec-cb-bd-de-ef-fb-ba-ae

Örnek (Euler çizgesi değil)



Hamilton Çizgeleri

Tanım

Hamilton çizgesi: kapalı, kapsayan bir yol düzenlenebilen çizge

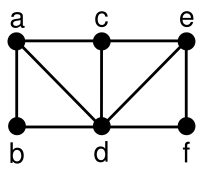
yol: dugumler yenilenmeyecek

tüm düğümleri ziyaret et ve başla dön tekrarsız şekilde

kapsayan yol: cizgedeki butun dugumlere ugrayan yol

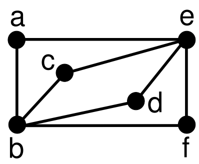
Hamilton Çizgesi Örnekleri

Örnek (Hamilton çizgesi)



ac-ce-ef-fd-db-ba

Örnek (Hamilton çizgesi değil)



e'ye dikkat

tekrar var

Bağlantı Matrisi Connectivity Matrix

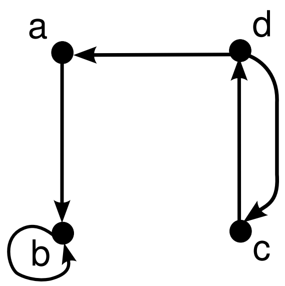
- çizgenin bitişiklik matrisi A ise
  A^k matrisinin (i,j) elemanı i. düğüm ile j. düğüm arasındaki
  k uzunluklu dolaşıların sayısını gösterir
- n düğümlü yönsüz bir çizgede iki düğüm arasındaki uzaklık en fazla $n-1$ olabilir
- bağlantı matrisi:
  $C = A^1 + A^2 + A^3 + \cdots + A^{n-1}$
  - bütün elemanlar sıfırdan farklı ise çizge bağlıdır

Warshall Algoritması

- düğümler arasındaki dolaşıların sayısı yerine dolaşı olup olmadığını belirlemek daha kolay
- sırayla her düğüm için:
  - o düğüme gelinebilen düğümlerden (matriste o sütunda 1 olan satırlardan)
  - o düğümden gidilebilen düğümlere (matriste o satırda 1 olan sütunlara)

Warshall Algoritması Örneği

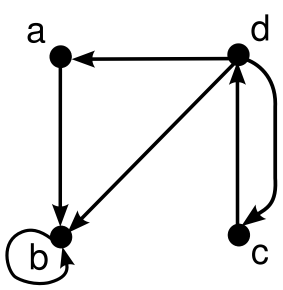
Örnek



	a	b	c	d
a	0	1	0	0
b	0	1	0	0
c	0	0	0	1
d	1	0	1	0

Warshall Algoritması Örneği

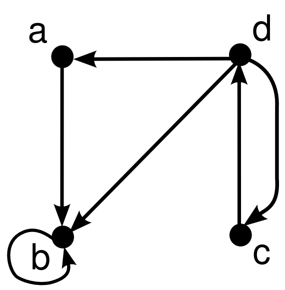
Örnek



	a	b	c	d
a	0	1	0	0
b	0	1	0	0
c	0	0	0	1
d	1	1	1	0

Warshall Algoritması Örneği

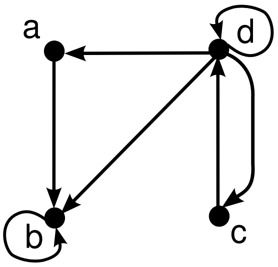
Örnek



	a	b	c	d
a	0	1	0	0
b	0	1	0	0
c	0	0	0	1
d	1	1	1	0

Warshall Algoritması Örneği

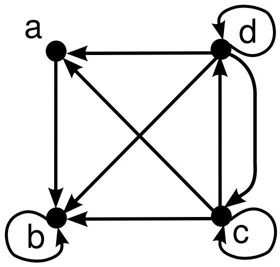
Örnek



	a	b	c	d
a	0	1	0	0
b	0	1	0	0
c	0	0	0	1
d	1	1	1	1

Warshall Algoritması Örneği

Örnek



	a	b	c	d
a	0	1	0	0
b	0	1	0	0
c	1	1	1	1
d	1	1	1	1

Düzlemsel Çizgeler

planar graph

Tanım

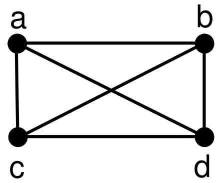
Ayrıtları kesişmeyecek şekilde bir düzleme çizilebilen bir çizge düzlemseldir.

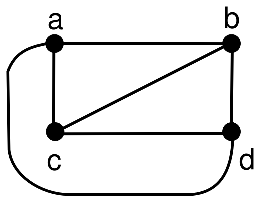
- G çizgesinin bir haritası: G çizgesinin düzlemsel bir çizimi

Düzlemsel Çizge Örneği

Örnek (K_4) değil

planar





Bölgeler

- bir harita düzlemi bölgelere ayırır
- bir bölgenin *kertesi*: bölgeyi çevreleyen kapalı gezinin uzunluğu

Teorem

r_i bölgesinin kertesi d_{r_i} olsun bölgedeki kenar sayısı

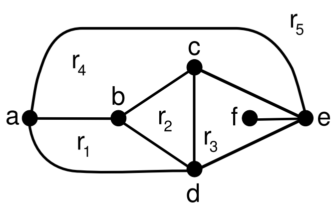
$$|E| = \frac{\sum_i d_{r_i}}{2}$$

Bir bölge (face), grafın çiziminde kenarlarla çevrili en küçük kapalı alan olarak ayırt edilir; kenarların içinden geçmeden bir noktadan diğerine gidilemiyorsa, orası ayrı bir bölgedir.

Bölge, kenarların ayırdığı en küçük alan olup; köprü varsa sınırında kenar tekrarı görülebilir.

Bölge Örneği

Örnek



$d_{r_1} = 3$ (abda)
$d_{r_2} = 3$ (bcdb)
$d_{r_3} = 5$ (cdefec)
$d_{r_4} = 4$ (abcea)
$d_{r_5} = 3$ (adea) dış bölge

$\sum_r d_r = 18$
$|E| = 9$

Planar graf teorisinde bölgeler (faces) birbirini kapsamaz; yalnızca kenar ve düğümlerle sınır komşusu olabilirler.

https://en.wikipedia.org/wiki/Dual_graph

Euler Formülü

https://www.youtube.com/watch?v=-9OUyo8NFZg

Teorem (Euler Formülü)

$G = (V, E)$ bağlı, düzlemsel bir çizge olsun ve R bu çizgenin bir haritasındaki bölgeler kümesi olsun:

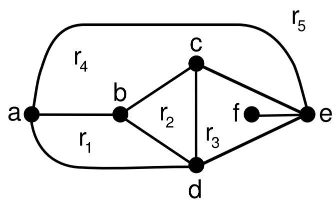
$$|V| - |E| + |R| = 2$$

dış bölgeyi sayarsan 2
saymazsan 1

V+ R- E düzgün olmayan şekillerde =1,
dış alanla 2 olur

Euler Formülü Örneği

Örnek



- $|V| = 6$, $|E| = 9$, $|R| = 5$

Düzlemsel Çizge Teoremleri

çizerek goster 66. slayt

Teorem

$G = (V, E)$ bağlı, düzlemsel bir çizge olsun ve $|V| \geq 3$ olsun:
$|E| \leq 3|V| - 6$

her bölgenin en az 3 kenarı vardır
varsayımı

Tanıt.

- bölge kertelerinin toplamı: $2|E|$
- bir bölgenin kertesi en az 3
  $\Rightarrow 2|E| \geq 3|R| \Rightarrow |R| \leq \frac{2}{3}|E|$
- $|V| - |E| + |R| = 2$
  $\Rightarrow |V| - |E| + \frac{2}{3}|E| \geq 2 \Rightarrow |V| - \frac{1}{3}|E| \geq 2$
  $\Rightarrow 3|V| - |E| \geq 6 \Rightarrow |E| \leq 3|V| - 6$

□

2 kenarlı bölge mümkündür,ama
bu teorem onları baştan dışarıda bırakır.

Düzlemsel Çizge Teoremleri

Teorem

$G = (V, E)$ bağlı, düzlemsel bir çizge olsun ve $|V| \geq 3$ olsun:
$\exists v \in V \ d_v \leq 5$

Tanıt.

- $\forall v \in V \ d_v \geq 6$ olsun
  $\Rightarrow 2|E| \geq 6|V|$
  $\Rightarrow |E| \geq 3|V|$
  $\Rightarrow |E| > 3|V| - 6$

celiski tabanlı kabul

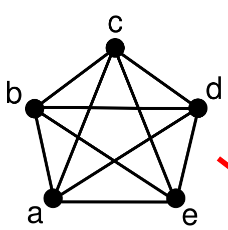
□

sağlanabilmesi için bazı dv'lerin 5'e eşit
veya küçük olması gerek.
Contradiction based proof

Düzlemsel Olmayan Çizgeler

Teorem

K_5 düzlemsel değildir.



E=<3|V|-6

Tanıt.

- $|V| = 5$
- $3|V| - 6 = 3 \cdot 5 - 6 = 9$
- $|E| \leq 9$ olmalı
- ama $|E| = 10$

□

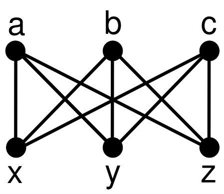
düzlemsel değil çakışan kenarlar var

Düzlemsel Olmayan Çizgeler

PLANAR= DUZLEMSEL

Teorem

$K_{3,3}$ düzlemsel değildir.



Tanıt.

- $|V| = 6, |E| = 9$
- düzlemsel ise $|R| = 5$ olmalı
- bir bölgenin kertesi en az 4
  $\Rightarrow \sum_{r \in R} d_r \geq 20$
- $|E| \geq 10$ olmalı
- ama $|E| = 9$

□

en az 4 düğümle bir alan çıkabilir , neden ?
paralellik ?

2 alt, 2 üst düğüm ile circuit

Kuratowski Teoremi

Teorem

G'nin K_5 ya da $K_{3,3}$'e homeomorfik bir altçizgesi var.

$\Leftrightarrow$

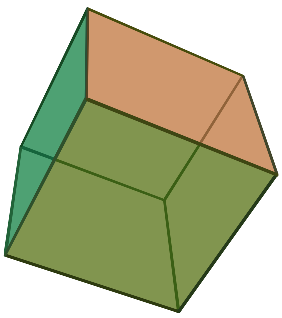
G düzlemsel değil.

Platon Cisimleri

- *düzgün çokyüzlü*: bütün yüzleri birbirinin eşi düzgün çokgenler olan üç boyutlu cisim
- bir düzgün çokyüzlünün iki boyutlu düzleme izdüşümü düzlemsel bir çizgedir
  - her köşe bir düğüm
  - her kenar bir ayrıt
  - her yüz bir bölge

Platon Cisimleri

Örnek (küp)



Platon Cisimleri

- v: düğüm (köşe) sayısı 8 köşe sayısı
- e: ayrıt (kenar) sayısı 12
- r: bölge (yüz) sayısı 6
- n: bir köşede birleşen yüz sayısı (düğüm kertesi) 3
- m: bir yüzü çevreleyen ayrıt sayısı (bölge kertesi) 4

- $m, n \geq 3$ n=3, m=4
- $2e = n \cdot v$ 2.12= 3.8
- $2e = m \cdot r$ 2.12= 4.6

Platon Cisimleri

- Euler formülünden:

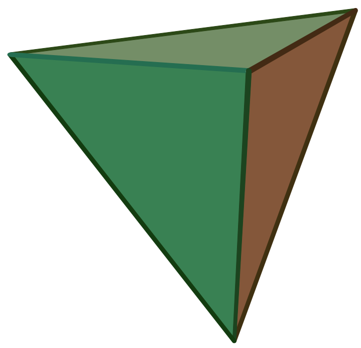
$$2 = v - e + r = \frac{2e}{n} - e + \frac{2e}{m} = e\left(\frac{2m - mn + 2n}{mn}\right) > 0$$

- $e, m, n > 0$:

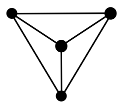
$$2m - mn + 2n > 0 \Rightarrow mn - 2m - 2n < 0$$
$$\Rightarrow mn - 2m - 2n + 4 < 4 \Rightarrow (m-2)(n-2) < 4$$

- bu eşitsizliği sağlayan değerler:
  1. $m = 3, n = 3$
  2. $m = 4, n = 3$
  3. $m = 3, n = 4$
  4. $m = 5, n = 3$
  5. $m = 3, n = 5$

Tetrahedron - Düzgün Dört Yüzlü



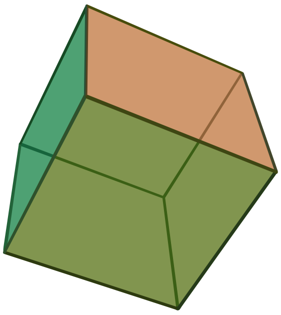
2.e=n.v=m*r
2*6= 4*3

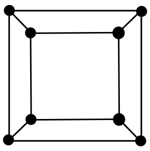


$m = 3, n = 3$

n: bir köşede birleşen yuz sayısı (dugum kertesi)
m: bir yuzü çevreleyen ayrıt sayısı (bolge kertesi)

Hexahedron - Düzgün Altı Yüzlü

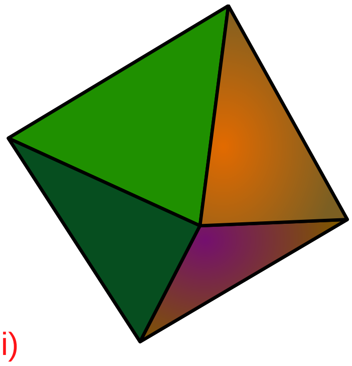


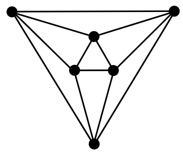


$m = 4, n = 3$

n: bir köşede birleşen yuz sayısı (dugum kertesi)
m: bir yuzü çevreleyen ayrıt sayısı (bolge kertesi)

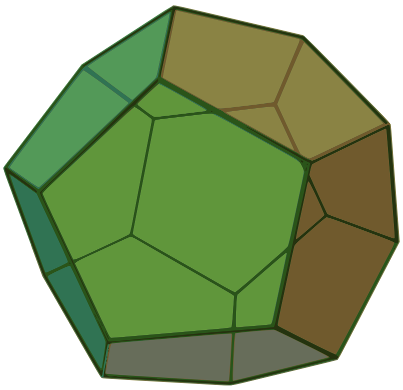
Octahedron - Düzgün Sekiz Yüzlü

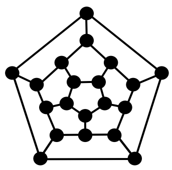




$m = 3, n = 4$

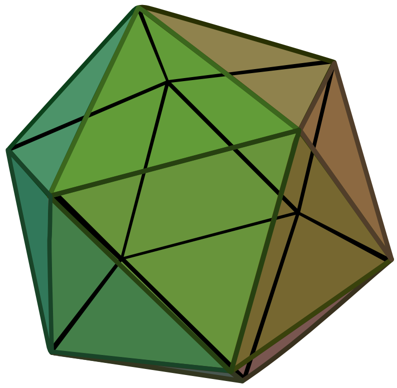
Dodecahedron - Düzgün Oniki Yüzlü





$m = 5, n = 3$

Icosahedron - Düzgün Yirmi Yüzlü



$m = 3, n = 5$

Çizge Boyama

Tanım
$G = (V, E)$ çizgesi için bir düzgün boyama: $f : V \to C$
C bir renk kümesi

- $\forall (v_i, v_j) \in E \; f(v_i) \neq f(v_j)$ f renk atama fonksiyonu
- $|C|$ en küçük olacak şekilde

en az renkle bu işi çözeceğiz

Çizge Boyama Örneği

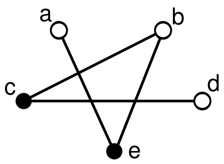
Örnek

- kimyasal maddeler üreten bir firma
- bazı maddeler birlikte tutulamıyor
- birbiriyle tutulamayan maddeler farklı alanlara depolanmalı
- en az sayıda depo alanı kullanılacak şekilde maddeleri depola

Çizge Boyama Örneği

Örnek

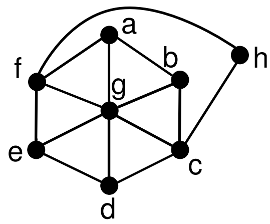
- her madde bir düğüm
- birlikte tutulamayan maddeler bitişik



2 renk

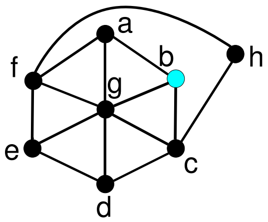
Çizge Boyama Örneği

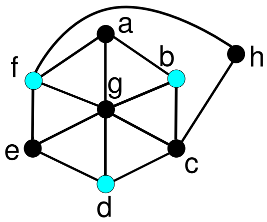
Örnek



Çizge Boyama Örneği

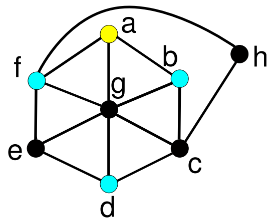
Örnek

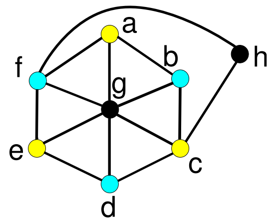




Çizge Boyama Örneği

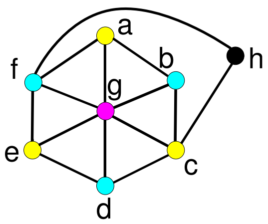
Örnek

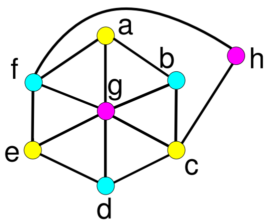




Çizge Boyama Örneği

Örnek





Kromatik Sayı

Tanım

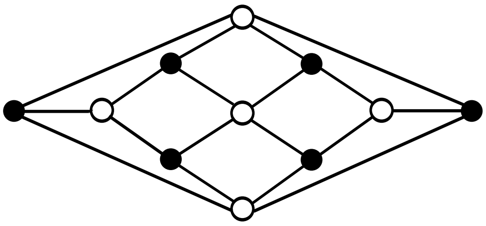
G çizgesinin kromatik sayısı: $\chi(G)$

- G çizgesini boyamak için gerekli en az renk sayısı

- $\chi(G)$'nin hesaplanması çok zor bir problem
- $\chi(K_n) = n$

Kromatik Sayı Örneği

Örnek (Herschel çizgesi)



- kromatik sayı: 2

Çizge Boyama Örneği

Örnek (Sudoku)

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

- her hücre bir düğüm
- aynı satırdaki hücreler bitişik
- aynı sütundaki hücreler bitişik
- aynı 3×3'lük bloktaki hücreler bitişik
- her rakam bir renk
- problem: kısmen boyalı bir çizgenin düzgün boyanması

Bölge Boyama

rübik küp

https://www.youtube.com/watch?v=9V-zpm9aj2A

- bir haritayı bitişik bölgelere farklı renkler atayacak şekilde boyama

Teorem (Dört Renk Teoremi)

Bir haritadaki bölgeleri boyamak için dört renk yeterlidir.

İç Anadolu: Sarı
Güneydoğu:Sarı
Doğu : Mavi
Marmara: Mavi
Ege: Siyah
Akdeniz, Karadeniz: Yeşil

Çizgelerde Arama

- $G = (V, E)$ çizgesinin düğümlerinin v_1 düğümünden başlanarak aranması
- derinlemesine
- enlemesine

Derinlemesine Arama

1. $v \leftarrow v_1$, $T = \emptyset$, $D = \{v_1\}$
2. $2 \le i \le |V|$ içinde $(v, v_i) \in E$ ve $v_i \notin D$ olacak şekilde en küçük i'yi bul
   - böyle bir i yoksa: 3. adıma git
   - varsa: $T = T \cup \{(v, v_i)\}$, $D = D \cup \{v_i\}$, $v \leftarrow v_i$, 2. adıma git
3. $v = v_1$ ise sonuç T
4. $v \neq v_1$ ise $v \leftarrow parent(v)$, 2. adıma git

komşular için for

Enlemesine Arama

1. $T = \emptyset$, $D = \{v_1\}$, $Q = (v_1)$
2. Q boş ise: sonuç T
3. Q boş değilse: $v \leftarrow front(Q)$, $Q \leftarrow Q - v$
   $2 \le i \le |V|$ için $(v, v_i) \in E$ ayrıtlarına bak:
   - $v_i \notin D$ ise: $Q = Q + v_i$, $T = T \cup \{(v, v_i)\}$, $D = D \cup \{v_i\}$
   - 3. adıma git

tüm komşular için for

Durum	DFS	BFS
Tüm çözümleri keşfetmek	✅	
Döngüleri tespit etmek	✅	
En kısa yolu bulmak (unweighted graf)		✅
Hafıza kısıtlaması olduğunda	✅	
Katmanlı yapıda arama		✅
Çözüm derinde olabilir	✅	
Çözüm yüzeyde veya yakınsa		✅

Kaynaklar

Okunacak: Grimaldi

- Chapter 11: An Introduction to Graph Theory
- Chapter 7: Relations: The Second Time Around
  - 7.2. Computer Recognition: Zero-One Matrices and Directed Graphs

Ağaç

kapalı yol olmamalı

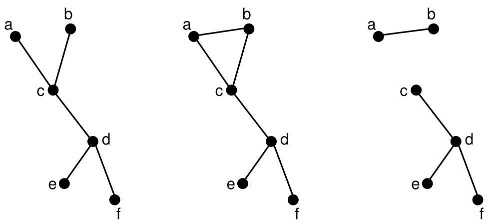
Tanım

ağaç: çevre içermeyen, bağlı çizge   cycle

- *orman*: bağlı bileşenleri ağaçlar olan çizge

Ağaç Örnekleri

Örnek

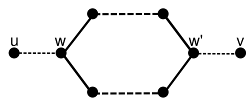


Ağaç Teoremleri

Teorem

Bir ağaçta herhangi iki farklı düğüm arasında bir ve yalnız bir yol vardır.

- ağaç bağlı olduğu için en az bir yol vardır
- birden fazla yol olsaydı çevre oluştururlardı



Ağaç Teoremleri

Teorem

$T = (V, E)$ *bir ağaç olsun:*

$$|E| = |V| - 1$$

- tanıt yöntemi: ayrıt sayısı üzerinden tümevarım

Ağaç Teoremleri

Tanıt: taban adımı

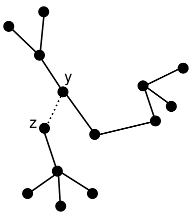
- $|E| = 0 \Rightarrow |V| = 1$
- $|E| = 1 \Rightarrow |V| = 2$
- $|E| = 2 \Rightarrow |V| = 3$
- $|E| \leq k$ için $|E| = |V| - 1$ varsayalım

Ağaç Teoremleri

Tanıt: tümevarım adımı.

- $|E| = k + 1$

V1=9, E1=8
V2=5, E2=4



- (y, z) ayrıtını çıkaralım:
  $T_1 = (V_1, E_1)$, $T_2 = (V_2, E_2)$

$$\begin{aligned} |V| &= |V_1| + |V_2| \\ &= |E_1| + 1 + |E_2| + 1 \\ &= (|E_1| + |E_2| + 1) + 1 \\ &= |E| + 1 \end{aligned}$$

z-y arsı ile bagli olsaydi

Ağaç Teoremleri

Teorem
Bir ağaçta kertesi 1 olan en az iki düğüm vardır.

Tanıt.

- $2|E| = \sum_{v \in V} d_v$
- kertesi 1 olan tek bir düğüm olduğunu varsayalım:
  $\Rightarrow 2|E| \geq 2(|V| - 1) + 1$
  $\Rightarrow 2|E| \geq 2|V| - 1$
  $\Rightarrow |E| \geq |V| - \frac{1}{2} > |V| - 1$

contradiction ile ispat

□

Ağaç Teoremleri

Teorem

T bir ağaçtır (bağlıdır ve çevre içermez).
$\Leftrightarrow$
T'de her düğüm çifti arasında bir ve yalnız bir yol vardır.
$\Leftrightarrow$
T bağlıdır ama herhangi bir ayrıt çıkarılırsa artık bağlı olmaz.
$\Leftrightarrow$
T çevre içermez ama herhangi iki düğüm arasına bir ayrıt eklenirse bir ve yalnız bir çevre oluşur.

Ağaç Teoremleri

Teorem

T bir ağaçtır (T bağlıdır ve çevre içermez.)
$\Leftrightarrow$
T bağlıdır ve $|E| = |V| - 1$.
$\Leftrightarrow$
T çevre içermez ve $|E| = |V| - 1$.

Köklü Ağaç

- düğümler arasında hiyerarşi tanımlanır
- hiyerarşi ayrıtlara doğal bir yön verir
  $\Rightarrow$ giriş ve çıkış kerteleri
- giriş kertesi 0 olan düğüm: kök
- çıkış kertesi 0 olan düğümler: yaprak
- yaprak olmayan düğümler: içdüğüm

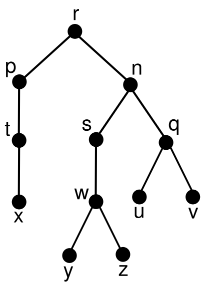
Düğüm Düzeyleri

Tanım
bir düğümün düzeyi: düğümün köke olan uzaklığı

- *anne*: bir üst düzeydeki bitişik düğüm
- *çocuk*: bir alt düzeydeki bitişik düğümler
- *kardeş*: aynı annenin çocuğu olan düğümler

Köklü Ağaç Örneği

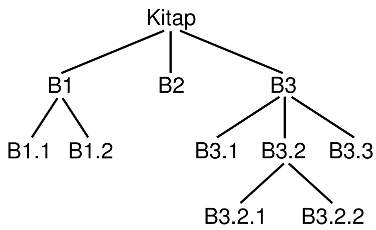
Örnek



- kök: r
- yapraklar: $x\ y\ z\ u\ v$
- içdüğümler: $r\ p\ n\ t\ s\ q\ w$
- y düğümünün annesi: w
  w düğümünün çocukları: y ve z
- y ve z kardeş

Köklü Ağaç Örneği

Örnek



Kitap
- B1
  - B1.1
  - B1.2
- B2
- B3
  - B3.1
  - B3.2
    - B3.2.1
    - B3.2.2
  - B3.3

Sıralı Köklü Ağaç

- kardeş düğümler soldan sağa doğru sıralanır
- evrensel adresleme sistemi
  - köke 0 adresini ver
  - 1. düzeydeki düğümlere soldan sağa doğru sırayla $1, 2, 3, \ldots$ adreslerini ver
  - v düğümünün adresi a ise, v düğümünün çocuklarına soldan sağa doğru sırayla $a.1, a.2, a.3, \ldots$ adreslerini ver

Sözlük Sırası

Tanım
b ve c iki adres olsun.
b'nin c'den önce gelmesi için aşağıdakilerden biri sağlanmalı:

1. $b = a_1 a_2 \ldots a_m x_1 \ldots$
   $c = a_1 a_2 \ldots a_m x_2 \ldots$
   x_1 x_2'den önce gelir

   1.2 ve 1.3 gibi

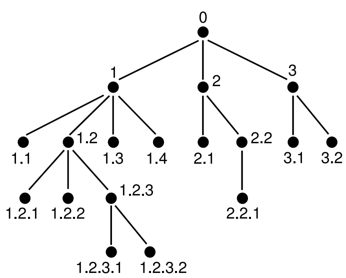
2. $b = a_1 a_2 \ldots a_m$
   $c = a_1 a_2 \ldots a_m a_{m+1} \ldots$

   1.2 ve 1.2.1 gibi

örnek

Sözlük Sırası Örneği

Örnek



- 0 - 1 - 1.1 - 1.2
  - 1.2.1 - 1.2.2 - 1.2.3
  - 1.2.3.1 - 1.2.3.2
  - 1.3 - 1.4 - 2
  - 2.1 - 2.2 - 2.2.1
  - 3 - 3.1 - 3.2

İkili Ağaçlar

ikili ağaçtaki düğüm kerteleri. 0 veya 2 ise tam ikili

çıktı dereceleri

Tanım
$T = (V, E)$ bir ikili ağaç: $\forall v \in V\ d_v^{o} \in \{0, 1, 2\}$

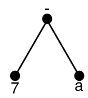
$T = (V, E)$ bir *tam* ikili ağaç: $\forall v \in V\ d_v^{o} \in \{0, 2\}$

İşlem Ağacı

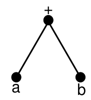
- bir ikili işlem bir ikili ağaçla temsil edilebilir
  - kökte işleç, çocuklarda işlenenler
- her matematiksel ifade bir ağaçla temsil edilebilir
  - içdüğümlerde işleçler, yapraklarda değişkenler ve değerler

İşlem Ağacı Örnekleri

Örnek $(7 - a)$

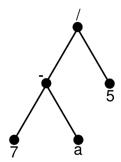


Örnek $(a + b)$

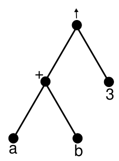


İşlem Ağacı Örnekleri

Örnek $((7 - a)/5)$

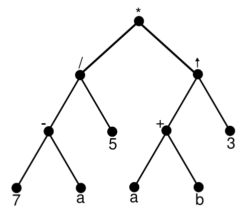


Örnek $((a + b) \uparrow 3)$



İşlem Ağacı Örnekleri

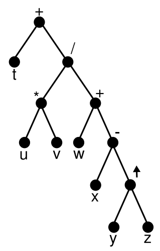
Örnek $(((7 - a)/5) * ((a + b) \uparrow 3))$



İşlem Ağacı Örnekleri

sol kök sağ

Örnek $(t + (u * v)/(w + x - y \uparrow z))$



İşlem Ağacında Geçişler

Infix, Prefix, Postfix Notation

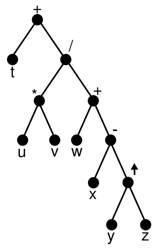
1. içek geçişi: sol altağacı tara, köke uğra, sağ altağacı tara
2. önek geçişi: köke uğra, sol altağacı tara, sağ altağacı tara
3. sonek geçişi: sol altağacı tara, sağ altağacı tara, köke uğra
   - *ters Polonyalı gösterilimi*

aynı işlem ağacını temsil ediyorlarsa infix, prefix ve postfix ifadeler her zaman aynı sonucu verir.

İçek Geçişi Örneği

sol kök sağ

Örnek

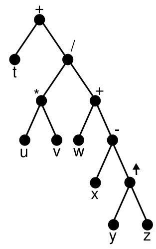


$t + u * v / w + x - y \uparrow z$

Önek Geçişi Örneği

kök sol sağ

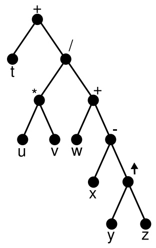
Örnek



$+ \, t \, / * u \, v + w - x \uparrow y \, z$

Sonek Geçişi Örneği

Örnek



$t \, u \, v * w \, x \, y \, z \uparrow - + / +$

İşlem Ağacının Değerlendirilmesi

- içek geçişinde öncelik için parantez gerekir
- önek ve sonek geçişlerinde parantez gerekmez

Sonek Değerlendirme Örneği

prefix infix postfix examples

Örnek ($t \, u \, v * w \, x \, y \, z \uparrow - + / +$)

$4 \, 2 \, 3 * 1 \, 9 \, 2 \, 3 \uparrow - + / +$

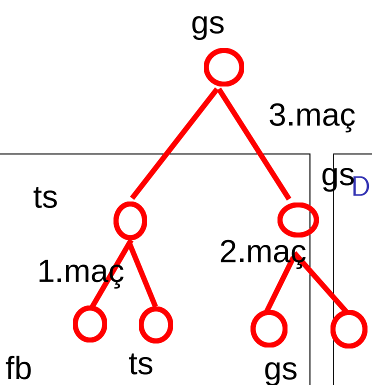
$$\begin{array}{lllllll}
4 & 2 & 3 & * & & & \\
4 & 6 & 1 & 9 & 2 & 3 & \uparrow \\
4 & 6 & 1 & 9 & 8 & - & \\
4 & 6 & 1 & 1 & + & & \\
4 & 6 & 2 & / & & & \\
4 & 3 & + & & & & \\
7 & & & & & &
\end{array}$$

Düzenli Ağaç

Tanım

$T = (V, E)$ bir m-li ağaç: $\forall v \in V \; d_v{}^o \leq m$

$T = (V, E)$ bir tam m-li ağaç: $\forall v \in V \; d_v{}^o \in \{0, m\}$



Düzenli Ağaç Teoremi

Teorem
$T = (V, E)$ bir tam m'li ağaç olsun.

- *n: düğüm sayısı*
- *l: yaprak sayısı*
- *i: içdüğüm sayısı*

O halde:

- $n = m \cdot i + 1$
- $l = n - i = m \cdot i + 1 - i = (m-1) \cdot i + 1$

m= her maç için düğüm sayısı

$$i = \frac{l-1}{m-1}$$

n=2*3+1 (kök iç düğüm)
l=7-3=4 yaprak sayısı

i= (4-1)/(2-1)=3

Düzenli Ağaç Örnekleri

Örnek

- 27 oyuncunun katıldığı bir tenis turnuvasında kaç maç oynanır?

- her oyuncu bir yaprak: $l = 27$
- her maç bir içdüğüm: $m = 2$
- maç sayısı: $i = \frac{l-1}{m-1} = \frac{27-1}{2-1} = 26$

2 için
l=i+1 -- i=7

i= (8-1) /(2-1)=7

Düzenli Ağaç Örnekleri

(n-1)*3 +4=25
n=8

Örnek

- 25 adet elektrikli aygıtı 4'lü uzatmalarla tek bir prize bağlamak için kaç uzatma gerekir?

- her aygıt bir yaprak: $l = 25$
- her uzatma bir içdüğüm: $m = 4$
- uzatma sayısı: $i = \frac{l-1}{m-1} = \frac{25-1}{4-1} = 8$

Karar Ağaçları

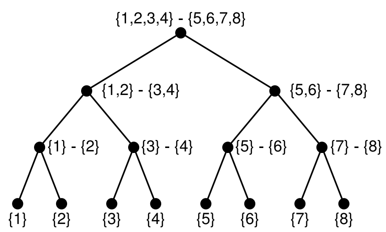
Örnek

- 8 madeni paranın biri sahte (daha ağır)
- bir teraziyle sahtenin hangisi olduğu bulunacak

Karar Ağaçları

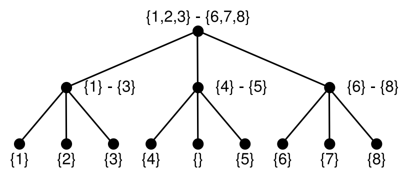
kollu terazi

Örnek (3 tartmada bulma)



Karar Ağaçları

Örnek (2 tartmada bulma)



Kaynaklar

Okunacak: Grimaldi

- Chapter 12: Trees
  - 12.1. Definitions and Examples
  - 12.2. Rooted Trees

Ağırlıklı Çizgeler

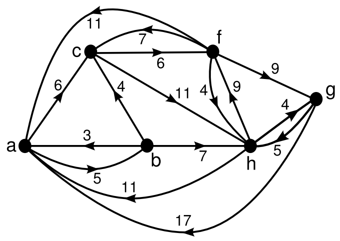
- ayrıtlara etiket atanabilir:
  ağırlık, uzunluk, maliyet, gecikme, olasılık, ...

En Kısa Yol

- bir düğümden bütün diğer düğümlere en kısa yolları bulma:
  Dijkstra algoritması

Dijkstra Algoritması Örneği

Örnek (başlangıç)

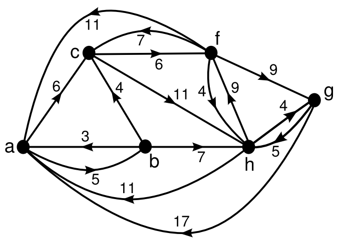


- başlangıç: c

a	$(\infty, -)$
b	$(\infty, -)$
c	$(0, -)$
f	$(\infty, -)$
g	$(\infty, -)$
h	$(\infty, -)$

Dijkstra Algoritması Örneği

Örnek (c düğümünden - taban uzaklık=0)



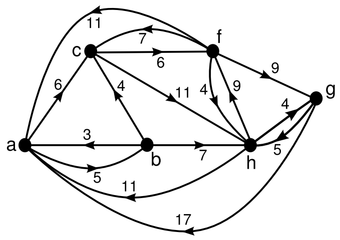
- $c \rightarrow f : 6, 6 < \infty$
- $c \rightarrow h : 11, 11 < \infty$

a	$(\infty, -)$	
b	$(\infty, -)$	
c	$(0, -)$	$\surd$
f	$(6, cf)$	
g	$(\infty, -)$	
h	$(11, ch)$	

- en yakın düğüm: f

Dijkstra Algoritması Örneği

Örnek (f düğümünden - taban uzaklık=6)



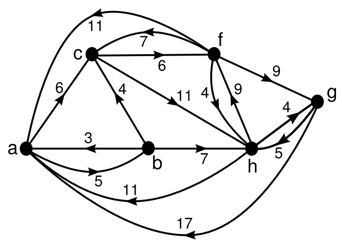
- $f \rightarrow a : 6 + 11, 17 < \infty$
- $f \rightarrow g : 6 + 9, 15 < \infty$
- $f \rightarrow h : 6 + 4, 10 < 11$

a	$(17, cfa)$	
b	$(\infty, -)$	
c	$(0, -)$	$\surd$
f	$(6, cf)$	$\surd$
g	$(15, cfg)$	
h	$(10, cfh)$	

- en yakın düğüm: h

Dijkstra Algoritması Örneği

Örnek (h düğümünden - taban uzaklık=10)



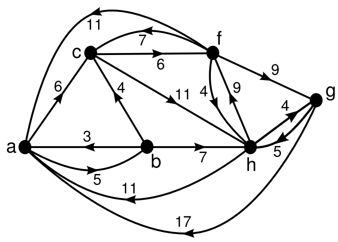
- $h \rightarrow a : 10 + 11, 21 \not< 17$
- $h \rightarrow g : 10 + 4, 14 < 15$

a	$(17, cfa)$	
b	$(\infty, -)$	
c	$(0, -)$	$\surd$
f	$(6, cf)$	$\surd$
g	$(14, cfhg)$	
h	$(10, cfh)$	$\surd$

- en yakın düğüm: g

Dijkstra Algoritması Örneği

Örnek (g düğümünden - taban uzaklık=14)



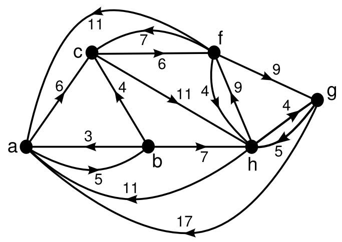
- $g \rightarrow a : 14 + 17, 31 \not< 17$

a	$(17, cfa)$	
b	$(\infty, -)$	
c	$(0, -)$	$\surd$
f	$(6, cf)$	$\surd$
g	$(14, cfhg)$	$\surd$
h	$(10, cfh)$	$\surd$

- en yakın düğüm: a

Dijkstra Algoritması Örneği

Örnek (a düğümünden - taban uzaklık=17)



- $a \rightarrow b : 17 + 5, 22 < \infty$

a	$(17, cfa)$	$\surd$
b	$(22, cfab)$	
c	$(0, -)$	$\surd$
f	$(6, cf)$	$\surd$
g	$(14, cfhg)$	$\surd$
h	$(10, cfh)$	$\surd$

- son düğüm: b

En Hafif Kapsayan Ağaç

Tanım
kapsayan ağaç:
çizgenin bütün düğümlerini içeren, ağaç özellikleri taşıyan bir altçizgesi

Tanım
en hafif kapsayan ağaç:
ayrıt ağırlıklarının toplamının en az olduğu kapsayan ağaç

Ayrıtların bir alt kümesini, tüm düğümleri kapsayacak ve ayrıtların toplam ağırlığını minimum yapacak şekilde bulur.

Kruskal Algoritması

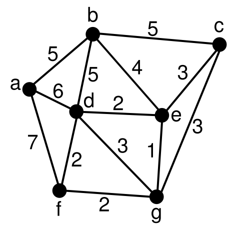
en kısa kenarlar üzerinden işlem

Kruskal algoritması weight

1. $i \leftarrow 1$, $e_1 \in E$, $wt(e_1)$ minimum 1 tane e seçmek 2 n çıkartmak demek
2. $1 \leq i \leq n - 2$ için:
   şu ana kadar seçilen ayrıtlar $e_1, e_2, \ldots, e_i$ ise kalan ayrıtlardan öyle bir e_{i+1} seç ki:
   - $wt(e_{i+1})$ minimum olsun
   - $e_1, e_2, \ldots, e_i, e_{i+1}$ altçizgesi çevre içermesin
3. $i \leftarrow i + 1$
   - if($i = n - 1 \Rightarrow e_1, e_2, \ldots, e_{n-1}$ ayrıtlarından oluşan G altçizgesi bir en hafif kapsayan ağaçtır
   - $i < n - 1 \Rightarrow$ 2. adıma git

Kruskal Algoritması Örneği

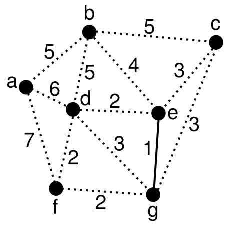
Örnek (başlangıç)



- $i \leftarrow 1$
- en düşük ağırlık: 1
  (e, g)
- $T = \{(e, g)\}$

Kruskal Algoritması Örneği

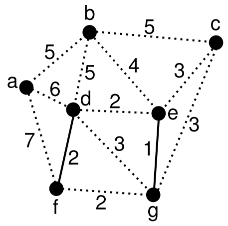
Örnek ($1 < 6$)



- en düşük ağırlık: 2
  $(d, e), (d, f), (f, g)$
- $T = \{(e, g), (d, f)\}$
- $i \leftarrow 2$

Kruskal Algoritması Örneği

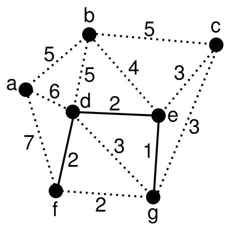
Örnek ($2 < 6$)



- en düşük ağırlık: 2
  $(d, e), (f, g)$
- $T = \{(e, g), (d, f), (d, e)\}$
- $i \leftarrow 3$

Kruskal Algoritması Örneği

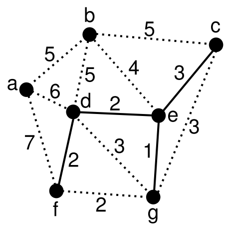
Örnek ($3 < 6$)



- en düşük ağırlık: 2
  (f, g) çevre oluşturuyor
- en düşük ağırlık: 3
  $(c, e), (c, g), (d, g)$
  (d, g) çevre oluşturuyor
- $T = \{(e, g), (d, f), (d, e), (c, e)\}$
- $i \leftarrow 4$

Kruskal Algoritması Örneği

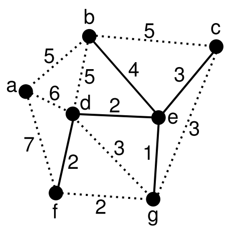
Örnek ($4 < 6$)



- $T = \{$
  $(e, g), (d, f), (d, e),$
  $(c, e), (b, e)$
  $\}$
- $i \leftarrow 5$

Kruskal Algoritması Örneği

Örnek ($5 < 6$)



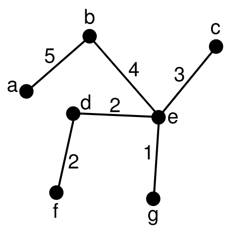
- $T = \{$
  $(e, g), (d, f), (d, e),$
  $(c, e), (b, e), (a, b)$
  $\}$
- $i \leftarrow 6$

i=n-1 ise bitti

7-1 ise bitir

Kruskal Algoritması Örneği

Örnek ($6 \not< 6$)



- toplam ağırlık: 17

Prim Algoritması

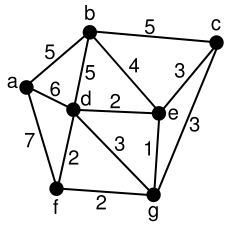
rastgele düğümle başla
olası en kısa yollarla devam et

Prim algoritması

1. $i \leftarrow 1,\ v_1 \in V,\ P = \{v_1\},\ N = V - \{v_1\},\ T = \emptyset$
2. $1 \leq i \leq n - 1$ için:
   $P = \{v_1, v_2, \ldots, v_i\},\ T = \{e_1, e_2, \ldots, e_{i-1}\},\ N = V - P$
   öyle bir $v_{i+1} \in N$ düğümü seç ki, bir $x \in P$ düğümü için
   $e = (x, v_{i+1}) \notin T,\ wt(e)$ minimum olsun
   $P \leftarrow P + \{v_{i+1}\},\ N \leftarrow N - \{v_{i+1}\},\ T \leftarrow T + \{e\}$
3. $i \leftarrow i + 1$
   - $i = n \Rightarrow e_1, e_2, \ldots, e_{n-1}$ ayrıtlarından oluşan G altçizgesi bir en hafif kapsayan ağaçtır
   - $i < n \Rightarrow$ 2. adıma git

Prim Algoritması Örneği

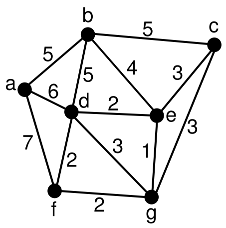
Örnek (başlangıç)



- $i \leftarrow 1$
- $P = \{a\}$
- $N = \{b, c, d, e, f, g\}$
- $T = \emptyset$

Prim Algoritması Örneği

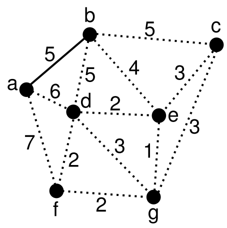
Örnek ($1 < 7$)



- $T = \{(a, b)\}$
- $P = \{a, b\}$
- $N = \{c, d, e, f, g\}$
- $i \leftarrow 2$

Prim Algoritması Örneği

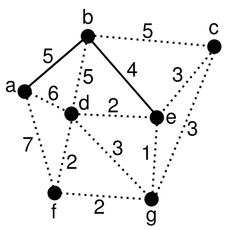
Örnek ($2 < 7$)



- $T = \{(a, b), (b, e)\}$
- $P = \{a, b, e\}$
- $N = \{c, d, f, g\}$
- $i \leftarrow 3$

Prim Algoritması Örneği

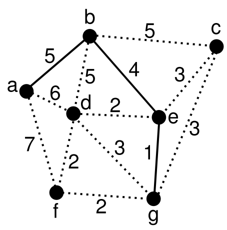
Örnek ($3 < 7$)



- $T = \{(a, b), (b, e), (e, g)\}$
- $P = \{a, b, e, g\}$
- $N = \{c, d, f\}$
- $i \leftarrow 4$

Prim Algoritması Örneği

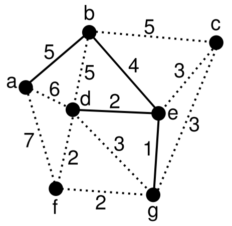
Örnek ($4 < 7$)



- $T = \{(a, b), (b, e), (e, g), (d, e)\}$
- $P = \{a, b, e, g, d\}$
- $N = \{c, f\}$
- $i \leftarrow 5$

Prim Algoritması Örneği

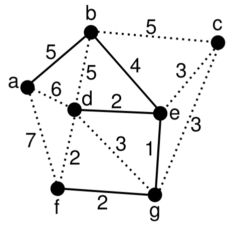
Örnek ($5 < 7$)



- $T = \{$
  $(a,b),(b,e),(e,g),$
  $(d,e),(f,g)$
  $\}$
- $P = \{a,b,e,g,d,f\}$
- $N = \{c\}$
- $i \leftarrow 6$

Prim Algoritması Örneği

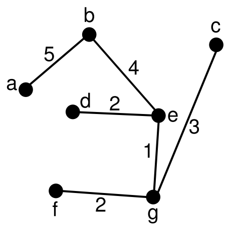
Örnek ($6 < 7$)



- $T = \{$
  $(a,b),(b,e),(e,g),$
  $(d,e),(f,g),(c,g)$
  $\}$
- $P = \{a,b,e,g,d,f,c\}$
- $N = \emptyset$
- $i \leftarrow 7$

Prim Algoritması Örneği

Örnek ($7 \not< 7$)



- toplam ağırlık: 17

Kaynaklar

Okunacak: Grimaldi

- Chapter 13: Optimization and Matching
  - 13.1. Dijkstra's Shortest Path Algorithm
  - 13.2. Minimal Spanning Trees:
    The Algorithms of Kruskal and Prim