Ayrık Matematik

Ağaçlar

H. Turgut Uyar   Ayşegül Gençata Yayımlı   Emre Harmancı

2001-2010

Lisans

Konular

1. Ağaçlar
   - Giriş
   - Köklü Ağaçlar
   - İkili Ağaçlar
   - Çizgelerde Arama
2. Özel Ağaçlar
   - Düzenli Ağaçlar
   - Karar Ağaçları
3. Ağaç Problemleri
   - En Hafif Kapsayan Ağaç

Ağaç

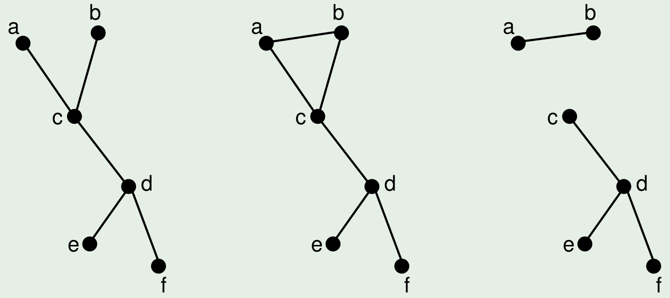
Tanım

ağaç: $T = (V, E)$
çevre içermeyen bağlı çizge

- bağlı bileşenleri ağaçlar olan çizge: *orman*

Ağaç Örnekleri

Örnek

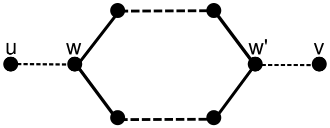


Ağaç Teoremleri

Teorem

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

- bağlı olduğu için bir yol var
- birden fazla yol olsaydı:



Ağaç Teoremleri

Teorem

$T = (V, E)$ *ağacında:* $|V| = |E| + 1$

- tanıt yöntemi: ayrıt sayısı üzerinden tümevarım

Ağaç Teoremleri

Tanıt: Taban adımı.

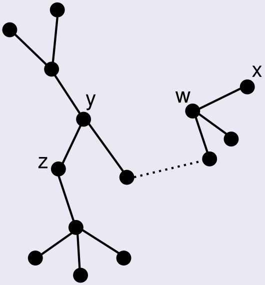
- $|E| = 0 \Rightarrow |V| = 1$
- $|E| = 1 \Rightarrow |V| = 2$
- $|E| = 2 \Rightarrow |V| = 3$

- $|E| \leq k$ için doğru olduğu varsayılsın

Ağaç Teoremleri

Tanıt: Tümevarım adımı.

- $|E| = k + 1$



- (y, z) çıkarılsın:
  $T_1 = (V_1, E_1),\ T_2 = (V_2, E_2)$

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

$\square$

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:
  $\Rightarrow 2|E| \geq 2(|V| - 1) + 1$
  $\Rightarrow 2|E| \geq 2|V| - 1$
  $\Rightarrow |E| \geq |V| - \frac{1}{2} > |V| - 1$ çelişki

□

Ağaç Teoremleri

Teorem

$T = (V, E) \wedge |V| \geq 2$ *ise aşağıdaki önermeler eşdeğerlidir:*

1. T *bir ağaçtır (bağlıdır ve çevre içermez)*
2. *her düğüm çifti arasında bir ve yalnız bir yol vardır*
3. T *bağlıdır ama herhangi bir ayrıt çıkarılırsa bu özelliğini yitirir*
4. T *çevre içermez ama herhangi iki düğüm arasına bir ayrıt eklenirse bu özelliğini yitirir*

- tanıt yöntemi: $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 1$

tanıtı atla

Ağaç Teoremleri

Tanıt: 1 $\Rightarrow$ 2.

T bağlıdır ve çevre içermez
$\Rightarrow$ her düğüm çifti arasında bir ve yalnız bir yol vardır

- varsayıma göre her düğüm çifti arasında bir yol vardır
- birden fazla yol olsaydı çevre olurdu

$\square$

Ağaç Teoremleri

Tanıt: $2 \Rightarrow 3$.

her düğüm çifti arasında bir ve yalnız bir yol vardır
$\Rightarrow$ T bağlıdır ama herhangi bir ayrıt çıkarılırsa bu özelliğini yitirir

- varsayıma göre yani her düğüm çifti arasında bir yol vardır yani T bağlıdır
- $e = (u, v)$ ayrıtını çizgeden çıkaralım:
  e ayrıtı u ile v düğümleri arasındaki tek yoldur, çıkarılırsa T bağlı olmaz

$\square$

Ağaç Teoremleri

Tanıt: $3 \Rightarrow 4$

T bağlıdır ama herhangi bir ayrıt çıkarılırsa bu özelliğini yitirir
$\Rightarrow$ T çevre içermez ama herhangi iki düğüm arasına bir ayrıt eklenirse bu özelliğini yitirir

1. T'nin çevre içermediğinin tanıtı
2. ayrıt eklemeyle çevre oluştuğunun tanıtı
3. ayrıt eklemeyle oluşan çevrenin tek olduğunun tanıtı

Ağaç Teoremleri

Tanıt: $3 \Rightarrow 4$

T bağlıdır ama herhangi bir ayrıt çıkarılırsa bu özelliğini yitirir $\Rightarrow$ T çevre içermez ama herhangi iki düğüm arasına bir ayrıt eklenirse bu özelliğini yitirir

Çevre içermediğinin tanıtı.

- T'de bir C çevresi olsun ve $e = (u, v) \in C$ olsun
- varsayıma göre T bağlı ama $T - e$ değil
  $\Rightarrow$ $T - e$ çizgesinde u ile v ayrı bileşenlerde
- oysa u ile v arasında $C - e$ yolu var

$\square$

Ağaç Teoremleri

Tanıt: $3 \Rightarrow 4$

T bağlıdır ama herhangi bir ayrıt çıkarılırsa bu özelliğini yitirir
$\Rightarrow$ T çevre içermez ama herhangi iki düğüm arasına bir ayrıt eklenirse bu özelliğini yitirir

Ayrıt eklemeyle çevre oluştuğunun tanıtı.

- T'ye $e = (u, v)$ ayrıtı eklensin
- varsayıma göre u ile v arasında bir yol var
  $\Rightarrow$ e ayrıtı ikinci bir yol oluşturur
  $\Rightarrow$ çevre oluşur

$\square$

Ağaç Teoremleri

Tanıt: $3 \Rightarrow 4$

T bağlıdır ama herhangi bir ayrıt çıkarılırsa bu özelliğini yitirir
$\Rightarrow$ T çevre içermez ama herhangi iki düğüm arasına bir ayrıt eklenirse bu özelliğini yitirir

Oluşan çevrenin tek olduğunun tanıtı.

- eklenen ayrıt: $e = (u, v)$
- oluşan çevre $C = P \cdot e$ çevresi olsun
- ikinci bir çevre daha oluştuğunu varsayalım: $C' = P' \cdot e$
  $\Rightarrow$ P ve P' yolları çevre oluşturur

$\square$

Ağaç Teoremleri

Tanıt: $4 \Rightarrow 1$.

T çevre içermez ama herhangi iki düğüm arasına bir ayrıt eklenirse bu özelliğini yitirir
$\Rightarrow$ T bağlıdır ve çevre içermez

- varsayıma göre T çevre içermez
- herhangi bir $e = (u, v)$ ayrıtı eklendiğinde çevre oluşuyor
  $\Rightarrow$ u ile v arasında yol var
  $\Rightarrow$ T bağlı

$\square$

Ağaç Teoremleri

Teorem

$T = (V, E) \wedge |V| \geq 2$ *ise aşağıdaki önermeler eşdeğerlidir:*

1. T *bir ağaçtır (bağlıdır ve çevre içermez)*
2. T *bağlıdır* $\wedge\ |E| = |V| - 1$
3. T *çevre içermez* $\wedge\ |E| = |V| - 1$

Köklü Ağaç

- düğümler arasında hiyerarşi
- ayrıtlarda doğal yön $\Rightarrow$ giriş ve çıkış kerteleri
  - giriş kertesi 0 olan (hiyerarşinin tepesindeki) düğüm: kök
  - çıkış kertesi 0 olan düğümler: yaprak
  - kök ve yapraklar dışında kalan düğümler: içdüğüm

Düğüm Düzeyleri

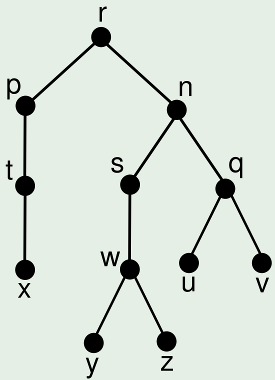
Tanım

düzey:
köke olan uzaklık

- anne: kökle arasındaki yolda kendisinden bir önceki düzeyde bulunan düğüm
- çocuk: bir sonraki düzeydeki komşu düğümler

Köklü Ağaç Örneği

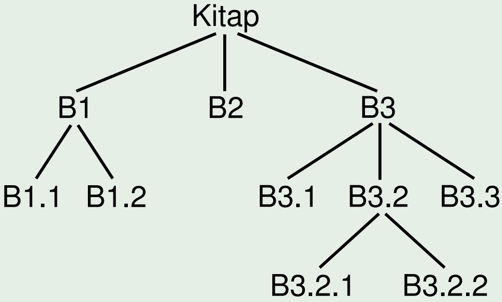
Örnek



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

Köklü Ağaç Örneği

Örnek (kitap düzeni)



Kitap

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

Sıralı Köklü Ağaç

- düğümler soldan sağa doğru sıralı
- 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

- b ve c iki adres olsun

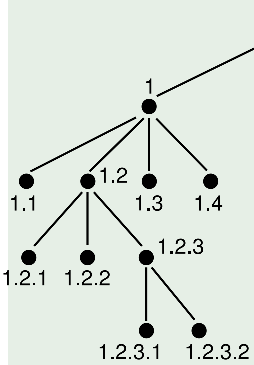
Tanım

$b < c$ olması için:

1. $b = a_1.a_2.\ldots.a_m$
   $c = a_1.a_2.\ldots.a_m.a_{m+1}\ldots a_n$
2. $b = a_1.a_2.\ldots.a_m.x_1\ldots y$
   $c = a_1.a_2.\ldots.a_m.x_2\ldots z$
   $x_1 < x_2$

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

Tanım

ikili ağaç:
$\forall v \in V \; {d_v}^o \in \{0, 1, 2\}$

Tanım

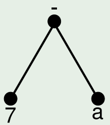
tam ikili ağaç:
$\forall v \in V \; {d_v}^o \in \{0, 2\}$

İşlem Ağacı

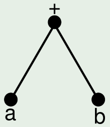
- ikili işlem tam ikili ağaçla temsil edilebilir
  - kökte işleç, çocuklarda işlenenler
- her işlem ikili ağaçla temsil edilebilir
  - içdüğümlerde işleçler, yapraklara değişkenler ve değerler
  - *tam ikili ağaç olmayabilir*

İşlem Ağacı Örnekleri

Örnek $(7 - a)$

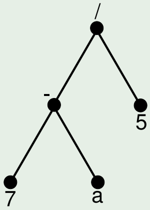


Örnek $(a + b)$

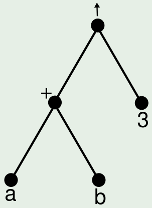


İşlem Ağacı Örnekleri

Örnek $((7 - a)/5)$

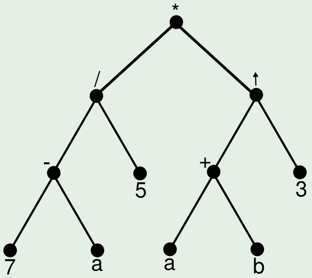


Örnek $((a + b) \uparrow 3)$



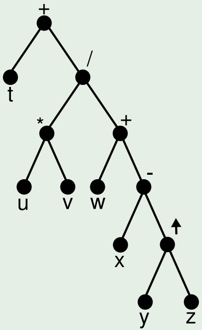
İşlem Ağacı Örnekleri

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



İşlem Ağacı Örnekleri

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

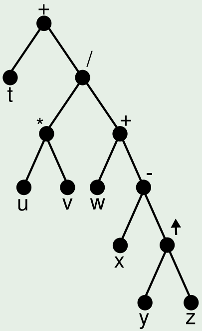


İşlem Ağacında Geçişler

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*

Önek Geçişi Örneği

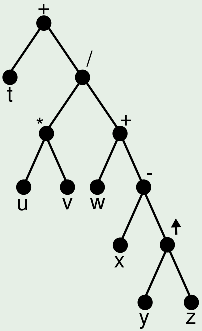
Örnek



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

İçek Geçişi Örneği

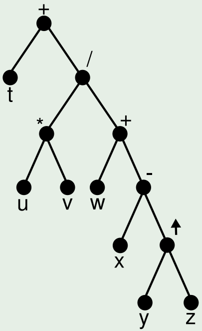
Örnek



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

Sonek Geçişi Örneği

Örnek



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

İşlem Ağacının Değerlendirilmesi

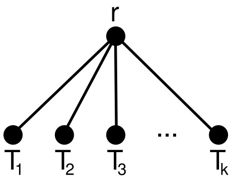
- işlem ağacında öncelikler:
  - içek geçişi parantez gerektirir
  - önek ve sonek parantez gerektirmez

İşlem Ağacı Değerlendirme Örneği

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

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

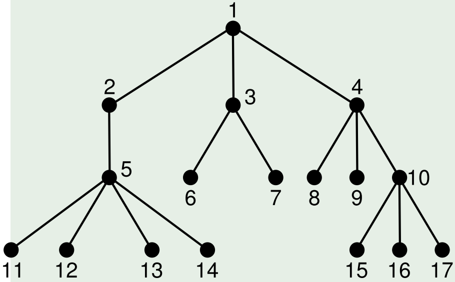
Sıralı Ağaçlar



- önek: $r, T_1, T_2, T_3, \ldots, T_k$
- sonek: $T_1, T_2, T_3, \ldots, T_k, r$

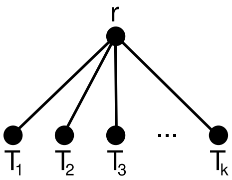
Sıralı Geçiş Örneği

Örnek



- önek:
  1, 2, 5, 11, 12, 13, 14, 3, 6, 7, 4, 8, 9, 10, 15, 16, 17
- sonek:
  11, 12, 13, 14, 5, 2, 6, 7, 3, 8, 9, 15, 16, 17, 10, 4, 1

Sıralı Ağaçlar

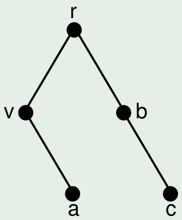


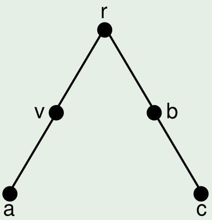
- içek: $T_1, r, T_2, T_3, \ldots, T_k$

Sıralı Ağaçlar

Örnek

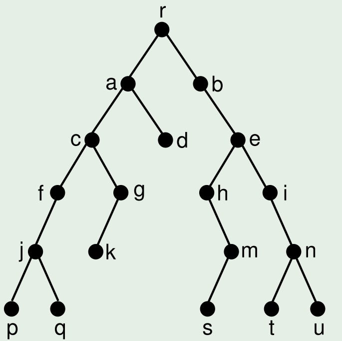
- aşağıdaki iki ağaç farklı





İçek Geçişi Örneği

Örnek



- p, j, q, f, c, k, g, a, d, r, b, h, s, m, e, i, t, n, u

Çizgelerde Arama

- $G = (V, E)$ çizgesinin v_1 düğümünü kök alarak kapsayan ağacının bulunması
  - derinlemesine
  - enlemesine

Derinlemesine Arama

1. $v \leftarrow v_1$, $T = \emptyset$, $D = \{v_1\}$
2. $2 \leq i \leq |V|$ 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

Enlemesine Arama

1. $T = \emptyset$, $D = \{v_1\}$, $Q = (v_1)$
2. Q boş ise: sonuç T
3. Q boş değilse: $v \leftarrow front(Q)$, $Q \leftarrow Q - v$
   $2 \leq i \leq |V|$ 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

Düzenli Ağaç

Tanım

m'li ağaç:
yapraklar dışındaki bütün düğümlerin çıkış kerteleri m

Düzenli Ağaç Teoremleri

Teorem

bir m'li ağaçta

- *düğüm sayısı* n
- *yaprak sayısı* l
- *içdüğüm sayısı (kök dahil)* i

ise

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

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

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: $i = \frac{l-1}{m-1} = \frac{27-1}{2-1} = 26$

Düzenli Ağaç Örnekleri

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: $i = \frac{l-1}{m-1} = \frac{25-1}{4-1} = 8$

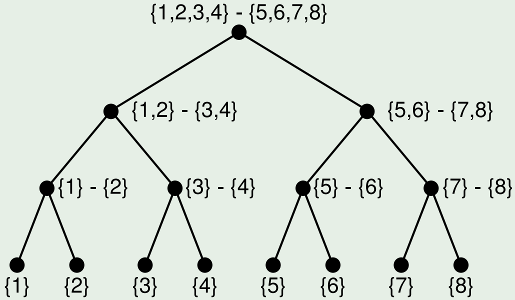
Karar Ağaçları

Örnek (sahte madeni para problemi)

- 8 madeni paranın biri sahte (daha ağır)
- bir teraziyle en az sayıda tartmayla sahteyi bulmak

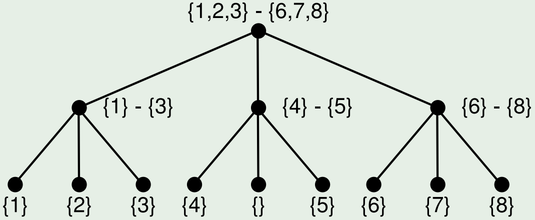
Karar Ağaçları

Örnek (3 tartmada bulma)



Karar Ağaçları

Örnek (2 tartmada bulma)



Kapsayan Ağaç

Tanım

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

Tanım

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

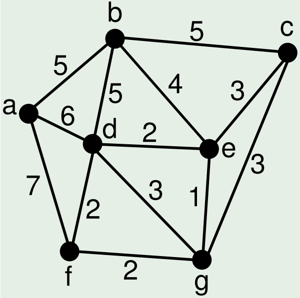
Kruskal Algoritması

Kruskal algoritması

1. $i \leftarrow 1$, $e_1 \in E$, $wt(e_1)$ minimum
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
   - $e_1, e_2, \ldots, e_i, e_{i+1}$ altçizgesi çevre içermiyor
3. $i \leftarrow i + 1$
   - $i = n - 1 \Rightarrow e_1, e_2, \ldots, e_{n-1}$ ayrıtlarından oluşan G altçizgesi bir en hafif kapsayan ağaçtır
   - $i < n - 1 \Rightarrow$ 2. adıma git

Kruskal Algoritması Örneği

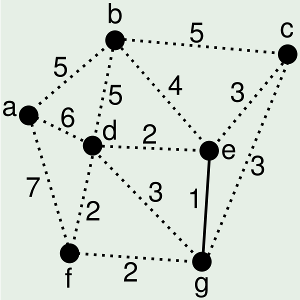
Örnek (başlangıç)



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

Kruskal Algoritması Örneği

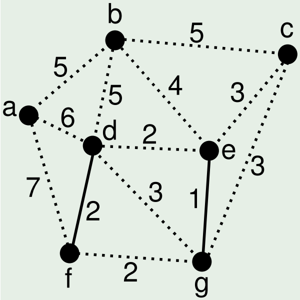
Örnek ($1 < 6$)



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

Kruskal Algoritması Örneği

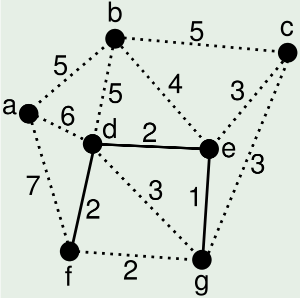
Örnek ($2 < 6$)



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

Kruskal Algoritması Örneği

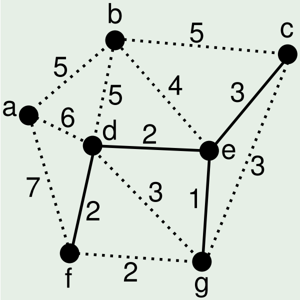
Örnek ($3 < 6$)



- 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

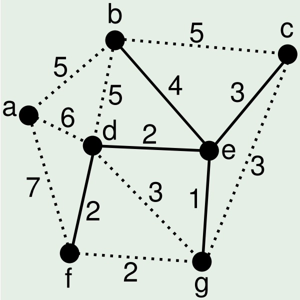
Örnek ($4 < 6$)



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

Kruskal Algoritması Örneği

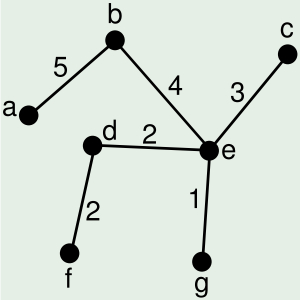
Örnek ($5 < 6$)



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

Kruskal Algoritması Örneği

Örnek ($6 \not< 6$)



- toplam ağırlık: 17

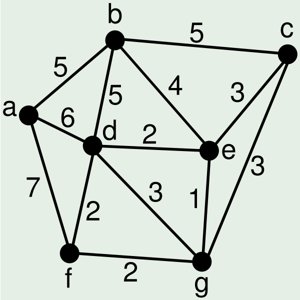
Prim Algoritması

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
   $P \leftarrow P + \{v_{i+1}\},\ N \leftarrow N - \{v_{i+1}\},\ T \leftarrow T + \{e\}$
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

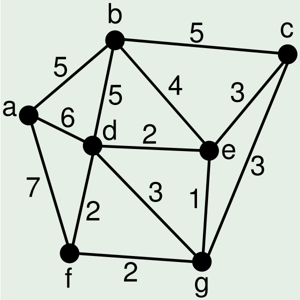
Örnek (başlangıç)



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

Prim Algoritması Örneği

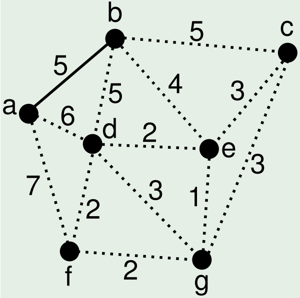
Örnek ($1 < 7$)



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

Prim Algoritması Örneği

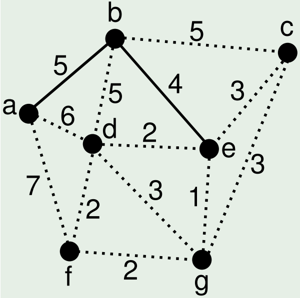
Örnek ($2 < 7$)



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

Prim Algoritması Örneği

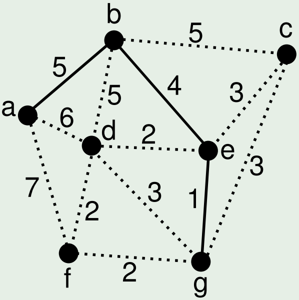
Örnek ($3 < 7$)



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

Prim Algoritması Örneği

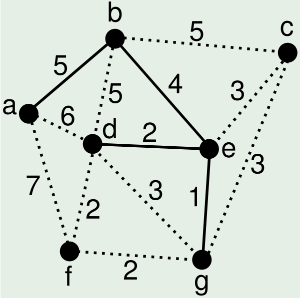
Örnek ($4 < 7$)



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

Prim Algoritması Örneği

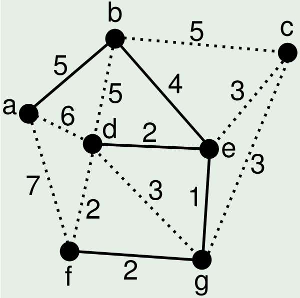
Örnek ($5 < 7$)



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

Prim Algoritması Örneği

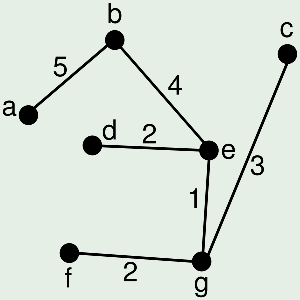
Örnek ($6 < 7$)



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

Prim Algoritması Örneği

Örnek ($7 \not< 7$)



- toplam ağırlık: 17

Kaynaklar

Okunacak: Grimaldi

- Chapter 12: Trees
  - 12.1. Definitions and Examples
  - 12.2. Rooted Trees
- Chapter 13: Optimization and Matching
  - 13.2. Minimal Spanning Trees: The Algorithms of Kruskal and Prim