DISKRETNA MATEMATIKA: STABLA

SADRŽAJ

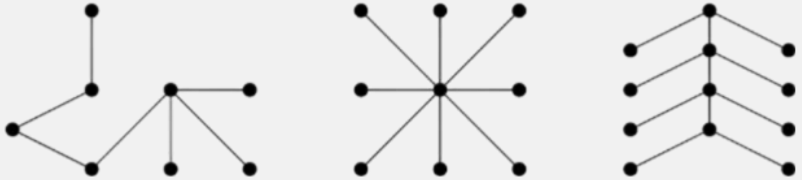
- Definicije stabla i šume
- Uslovi da graf bude stablo
- Priferov kod
- Definicija težinskog grafa
- Algoritmi...

Definicija stabla

Za prost graf G = (V,E) kažemo da je **stablo** ako vazi:

i. G je **povezan graf** i

ii. G je **acikličan graf**



<u>Teorema</u> „*Neka je* $G = (V,E)$ *i* $|V| = n \geq 2$. *Tada je* G *stablo ako i samo ako za svaka dva čvora* $u, v \in V$ *postoji jedinstven uv-put.*”

Dokaz

1. **Prva implikacija**: *Ako je G stablo, dokazati da postoji jedinstveni uv-put*.

- Pretpostavimo suprotno, tj. da G jeste stablo, ali ne postoji jedinstven uv-put izmedju nekih čvorova u i v.
- Posledice ove pretostavke:
- Ako postoji više od jednog puta izmedju u i v, to znači da postoji ciklus u G (jer se dva različita puta izmedju u i v zatvaraju u ciklus)
- Kontradikacija:
- Stablo je, po definiciji, graf koji je povezan i ne sadrži cikluse. Ako G ima ciklus onda nije stablo. Ovo je u suprotnosti sa pretpostavkom da je G stablo.
- Dakle, ako je G stablo, mora postojati jedinstven uv-put za bilo koja dva čvora u i v.

2. **Druga implikacija**: *Ako za svaka dva čvora u, v postoji jedinstven uv-put, dokazati da je G stablo.*

- Pretpostavimo suprotno, tj. Da za svaka dva čvora u, v ∈ V postoji jedinstven uv-put, ali G nije stablo.
- Posledice ove pretpostavke:
- Ako G nije stablo, onda:
- G nije povezan, ili
- G sadrži ciklus.
- Kontradikcija:
- U oba slučaja dolazimo do kontradikcije sa definicijom stabla. Dakle, ako za svaka dva čvora postoji jedinstven uv-put, G mora biti stablo

Lema

Neka je G = (V,E) stablo i neka je $|V| = n \geq 2$. Tada postoje bar dva čvora stepena 1.

Lema

Neka je $G = (V,E)$, $|V| = n \geq 2$, i neka je $d_G(u)=1$ za neki čvor $u \in V$. Tada je G stablo ako i samo ako je G u stablo.

Teorema „*Neka je* $G = (V,E)$ *i* $|V| = n \geq 2$. *Tada je* G *stablo ako i samo ako je* G *povezan graf i* $|E| = n-1$.”

Dokaz

(→) Prema definiciji stabla, G je povezan graf. Indukcijom po n ćemo pokazati da je |E| = n - 1

Baza n =2: Stablo sa dva čvora ima tačno jednu granu.

Induktivni korak T_{n-1} → Tn :

Ako je G stablo onda postoji čvor u sa osobinom $d_G(u)=1$. Graf G0 = G u ima osobinu

$$|V(G')| = |V(G)| -1 = n -1 \text{ i } |E(G')| = |E(G)| - 1.$$

Ako je G stablo, onda je prema *prethodnoj lemi* G' stablo. Prema induktivnoj pretpostavci je |E(G')| = n - 1, a odatle je |E(G)| = |E(G')| +1 = n.

(←) Indukcijom po n.

Baza n =2: Povezan graf sa dva čvora i jednom granom je stablo.

Induktivni korak T_{n-1} → Tn :

Ako je E(G) = V(G)−1, onda postoji čvor u sa osobinom dG(u) ≤ 1. Kako je G povezan, mora važiti $d_G(u)=1$ i graf G' u je povezan graf sa osobinom

$$|V(G')| = |V(G)| - 1 = n \text{ i } |E(G')| = |E(G)| - 1 = n - 1.$$

Prema induktivnoj pretpostavci je sada G' u stablo. Prema *prethodnoj lemi*, G je stablo.

Lema

Neka je $G = (V,E)$, gde je $|V| = n \geq 2$ i $|E| \geq n$. Neka su $V(G_1),\ldots V(G_l)$ komponente povezanosti grafa G sa $k_1,\ldots,k_l$ čvorova, respektivno. Tada postoji $i \in \{1,\ldots,l\}$ sa osobinom $|E(Gi)| \geq k_i$.

<u>Teorema</u> „*Neka je $G = (V,E)$, gde je $|V| = n \geq 2$ i $|E| \geq n$. Tada G sadrži konturu.*”

Dokaz

Razmatramo dva slučaja.

(i) G je povezan: ako G nema konturu, onda je stablo → G ima n 1 grana.

(ii) G nije povezan: neka su $G_1,\ldots,G_l$ komponente povezanosti grafa G:

(ii) $|V(G_1)| = k_1,\ldots,|V(G_l)| = k_l$ $k_1 + \ldots + k_l = n$.

Prema *prethodnoj lemi*, postoji $i \in \{1,\ldots,l\}$ sa osobinom $|E(G_i)|$ k_i. Ako G_i nema konturu, onda je G_i stablo i ima k_i-1 granu, što dovodi do kontradikcije. Znači, G_i ima konturu, a samim tim i G.

Teorema *„Neka je $G = (V,E)$ i $|V| = n \geq 2$. Tada je G stablo akko je G acikličan i dodavanjem grane se dobija graf koji sadrži konturu."*

Dokaz

($\rightarrow$) Ako je G stablo, onda je G acikličan graf po definiciji. Posmatrajmo proizvoljna dva čvora u, v sa osobinom $uv \notin E(G)$. Kako je G povezan, postoji uv-put u G. Dodavanjem grane uv dobijamo konturu u $G + uv$.

($\leftarrow$) Treba pokazati da je G povezan. Neka su u i v proizvoljni čvorovi iz V. Imamo dva slučaja:

(i) Ako je $uv \in E$, onda je to uv-put.

(ii) (ii) Ako $uv \notin E$, onda $G + uv$ sadrži konturu koja sadrži uv. Oduzimanjem sa konture grane uv dobijamo uv-put u G.

Karakterizacija stabla

Neka je G = (V,E) prost graf. Sledeća tvrđenja sa ekvivalentna:

(i) G je stablo.

(ii) Za svaka dva čvora u, v ∈ V(G) postoji jedinstven put od u do v.

(iii) G je povezan i |E(G)| = |V(G)| −1.

(iv) G je povezan i brisanjem proizvoljne grane dobija se nepovezan graf (tj. G je minimalan povezan graf).

(v) G je acikličan i dodavanjem grane se dobija graf koji sadrži konturu (tj. G je maksimalan acikličan graf).

Pokrivajuće stablo

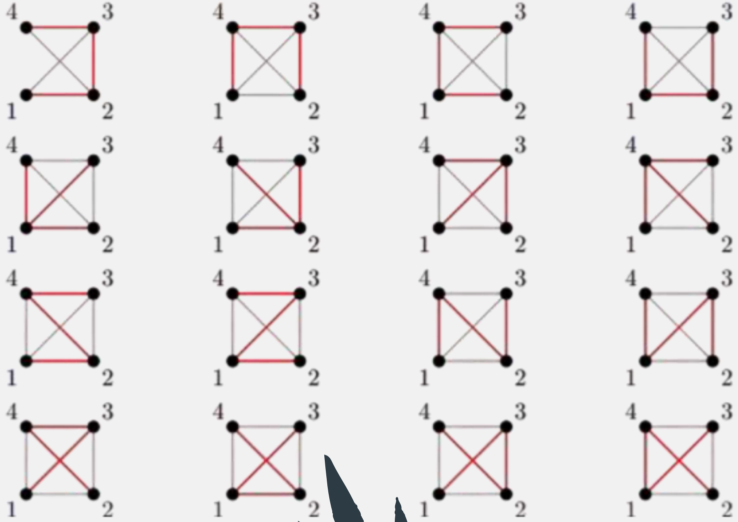
Kada se razmatraju problemi optimizacije na grafovima, često se dešava da optimalno rešenje ima ne-nula vrednosti samo na nekim podgrafovima koja su stabla i čiji skup čvorova je isti kao u plaznom grafu. Za takav podgraf kažemo da je pokrivajuće (ili razapinjuće ili razapeto) stablo.

Graf G1 je pokrivajuće stablo grafa G ako važe sledeće dve osobine:

(i) G1 je pokrivajući podgraf od G: $V(G1) = V(G)$ i $E(G1) \subseteq E(G)$;

(ii) G1 je stablo.

Zad. Koliko ima različitih pokrivajućih stabala grafa K4?



Teorema *„Graf G ima pokrivajuće stablo ako i samo ako je povezan."*

Dokaz

(→) Ako G ima pokrivajuće stablo, onda postoji put između svaka dva čvora stabla, a onda je to put i u grafu G.

(←) Neka je G povezan. Posmatraćemo dva slučaja.

1. $|V(G)| = 2$: Povezan graf sa dva čvora ima jednu granu i sopstveno je pokrivajuće stablo.

2. $|V(G)| = n \geq 3$: Za povezan graf važi da je $|E(G)| \geq n - 1$.

(a) Ako je $|E(G)| = n - 1$, povezan graf sa $n - 1$ grana je stablo. Znači, G je stablo, a ujedno i sopstveno pokrivajuće stablo.

(b) (b) Neka je $E(G)| = k \geq n$.

Priferov kod

Priferov kod za stabla je način predstavljanja stabla kao niza brojeva. Ovaj kod je posebno koristan za kodiranje stabala, jer svako stablo ima jedinstven Priferov kod.

Šta je Priferov kod?
Priferov kod je niz brojeva koji predstavlja označeno stablo sa nnn čvorova. Kod je dugačak n-2, jer se kodira samo n-2 uklanjanja čvorova iz stabla dok ne ostanu samo dva čvora.

<u>Teorema</u> „*Neka je $n \geq 2$. Broj različitih označenih stabala sa čvorovima $\{1, 2, \dots, n\}$ jednak je n^{n-2}.*”

Dokaz

Ako je n = 2, imamo jedno označeno stablo i tvrđenje važi. Posmatraćemo sada $n \geq 3$ i pokazaćemo dva podtvrđenja:

(i) *svakom stablu sa čvorovima {1,...,n} možemo na jedinstven način pridružiti Prüferov niz* $(p_1,\ldots,p_{n-2})$ *koji čine n-2 cela broja iz skupa {1,...,n} (koja se mogu ponavljati);*

(ii) *svaki niz* $(p_1,\ldots,p_{n-2})$ *sa osobinom* $\{p_1,\ldots,p_{n-2}) \subseteq \{1,\ldots, n\}$ *je Priferov niz nekog stabla sa n čvorova.*

(i) Niz ćemo formirati kao što je objašnjeno u nastavku.

1. Odrediti najmanju oznaku lista u stablu i za p_1 uzeti oznaku njemu susednog čvora. Oduzeti iz grafa list sa oznakom p_1 (i njemu incidentnu granu).
2. Ponavljati prvi korak, dok god ne ostanu samo dva čvora u stablu. Znači za pi, $2 \leq i \leq n-2$, uzeti oznaku suseda lista (u novodobijenom stablu) sa najmanjom oznakom.

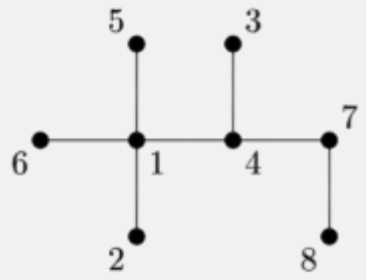
Tako smo svakom stablu pridružili Priferov niz.

Dokaz

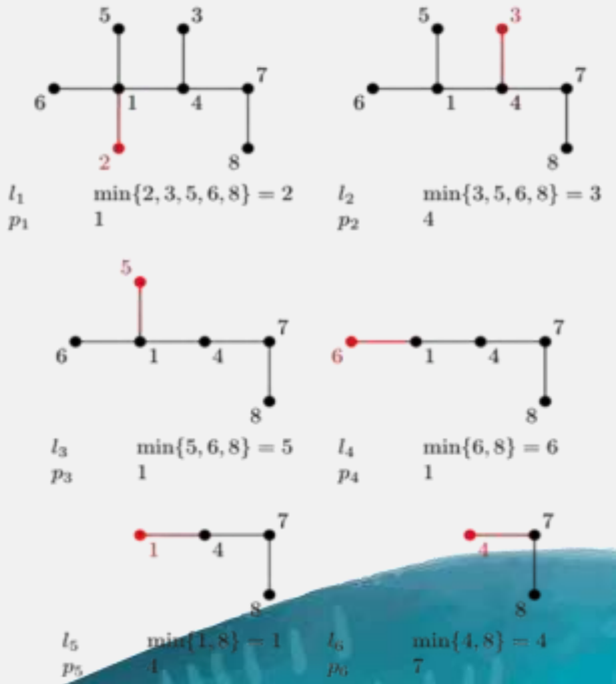
(ii) Neka je dat niz $(p_1,\ldots,p_{n-2})$. U nastavku ćem konstruisati stablo čiji je to Priferov niz.

1. Neka je l_1 najmanji broj koji se ne pojavljuje u skupu $\{p_1,\ldots,p_{n-2}\}$. To je morao biti list koji se skida u prvom koraku algoritma. Znači, treba spojiti granom čvorove p_1 i l_1.
2. U svakom narednom koraku, tražimo vrednost l_i koja će odgovarati najmanjoj oznaci lista koji skidamo kada formiramo niz. To je u svakom koraku najmanja vrednost iz skupa koji dobijamo kada iz $\{1,\ldots,n\}$ oduzmemo naredne članove niza (čim se pojavljuju u nizu, znači da nisu mogli biti skinuti kao listovi sa najmanjom oznakom) i prethodno skinute listove, tj. $l_i = \min((\{1,\ldots,n\}\setminus\{l_1,\ldots,l_{i1}\}) \setminus \{p_i,\ldots,p_{n2}\})$. Grana koju formiramo je $\{l_i, p_i\}$.
3. Preostala dva čvora, koji se nisu pojavili u skupu identifikovanih listova, povežemo granom.

Zadatak. Odrediti Priferov niz za stablo sa slike.

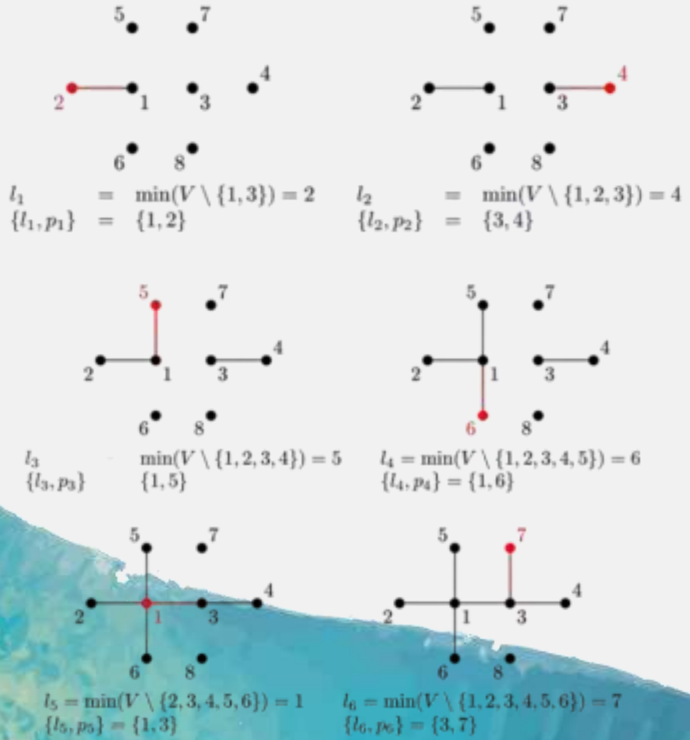


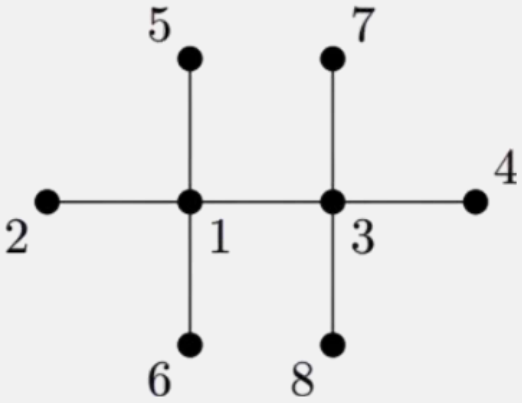
Rešenje. Priferov niz za dato stablo je (p1, p2, p3, p4, p5, p6) = (1, 4, 1, 1, 4, 7).



Zadatak. Odrediti stablo čiji je Priferov niz (1,3,1,1,3,3).

Rešenje. Prvo treba primetiti da je dužina niza n 2=6, odakle je broj čvorova stabla n = 8. Daćemo grafički prikaz formiranja stabla. Uvedimo oznaku V = {1, 2, 3, 4, 5, 6, 7, 8}.





Težinski grafovi

Grafički modeli koji se pojavljuju u praksi često zahtevaju dodeljivanje nekih **realnih vrednosti granama.**
Te vrednosti ćemo zvati **težinama grana**.

Težinski graf je uređena trojka (V,E,ω), gde je

$$\omega : E \to R$$

funkcija koja svakoj grani $e \in E$ dodeljuje realan broj (njenu težinu) $\omega(e)$.

Algoritmi za odredjivanje minimalnog pokrivajućeg stabla

Šta je minimalno pokrivajuće stablo (MST)?
Minimalno pokrivajuće stablo (MST) je podskup grana povezanog, neusmerenog grafa koji povezuje sve čvorove bez ciklusa i sa najmanjom ukupnom težinom.

1. Kruskalov algoritam
Istorija: Razvio ga je Joseph Kruskal 1956. godine.
Postupak:
Sortiraj sve grane po težini.
Iterativno dodaj grane sa najmanjom težinom koje ne formiraju ciklus (koristeći strukturu disjunktnih skupova – union-find).
Zaustavi se kada se doda n-1n-1n-1 grana (za nnn čvorova).
Karakteristike: Pogodan za grafove sa malim brojem grana (retki grafovi).

2. Primov algoritam
Istorija: Objavio ga je matematičar Vojtěch Jarník 1930, a redistribuirao ga je Robert Prim 1957. godine. Kasnije ga je popularizovao Edsger Dijkstra.
Postupak:
Počni sa proizvoljnim čvorom i dodaj ga u MST.
Nađi granu najmanje težine koja povezuje čvor u MST s čvorom izvan MST.
Ponavljaj dok svi čvorovi ne budu deo MST.
Karakteristike: Pogodan za guste grafove (mnogi čvorovi povezani).

```python
def dijkstra(graph, start):  1 usage
    """
    Funkcija za izračunavanje najkraćih puteva od početnog čvora do svih drugih čvorova.
    :param graph: Reprezentacija grafa kao rečnika gde su ključevi čvorovi,
                  a vrednosti liste suseda i njihovih težina [(sused, težina)].
    :param start: Početni čvor.
    :return: Dva rečnika:
             - distances: Najkraće udaljenosti od početnog čvora do svakog drugog čvora.
             - previous: Putanja koja vodi do svakog čvora.
    """
    # Inicijalizacija
    distances = {node: float('inf') for node in graph}
    distances[start] = 0
    previous = {node: None for node in graph}
    priority_queue = [(0, start)]  # (udaljenost, čvor)
    while priority_queue:
        current_distance, current_node = heapq.heappop(priority_queue)

        # Preskoči ako smo već pronašli kraći put
        if current_distance > distances[current_node]:
            continue
        # Proveri sve susede trenutnog čvora
        for neighbor, weight in graph[current_node]:
            distance = current_distance + weight

            # Ako smo našli kraći put do suseda, ažuriraj udaljenost i prethodnika
            if distance < distances[neighbor]:
                distances[neighbor] = distance
                previous[neighbor] = current_node
                heapq.heappush( *args: priority_queue, (distance, neighbor))
    return distances, previous
```

Hvala na pažnji!

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**