Spanning tree

Spanning tree grafu *G=(V,E)* je podgraf T⊆G takový, že V(T) = V(G) a T je strom

Alg: Jarnik, Borůvka,...

Minimum spanning tree

MST grafu G je spanning tree M takový, že w(M) ≤ w(T) pro všechny spanning trees T grafu G

Alg: Jarnik, Borůvka,...

Blue/ Red rule

Blue rule - pro zadaný řez s žádnými modrými hranami, vyber minimální neobarvenou crossing hranu a obarvi ji modře (CUT property)

Red rule - pro zadaný cyklus neobsahující žádnou červenou hranu, vyber maximální neobarvenou maximální hranu a obarvi ji červeně. (Cycle property)

Alg: Jarnik, Borůvka,...

Dense graph / Sparse graph

Dense graph je graf takový, kde počet hran se blíží maximálnímu počtu hran
Sparse graph je graf s pouze pár hranami

Kontext: grafy

Subforest / F-light a F-heavy edges

Subforest F grafy *G = (V, E)* je podgraf grafu G který je forest

F-light and heavy edges
Nechť F je subforest grafu *G = (V, E)* a *e* ∈ *E*.

- $W_F(e)$ - váha **nejtěžší hrany** na unikátní path v *F* mezi dvěma endpointy e
    - $w_F = \infty$ pokud e spojuje dvě různé komponenty z F
- $e \in E$ je **F-heavy** právě pokud $w(e) > w_F(e)$ a **F-light** v ostatních případech

Kontext: Minimum spanning tree
Alg: Karger-Klein-Tarjan algorithm

Arborescence

Pro **directed** graf $D = (V, E)$ a **root** $r \in V$ je **arborescence** (s kořenem v r) podgraf takové $T = (V, F)$, že:

- T je spanning tree (pokud nebereme v potaz orientaci hran)
- Existuje directed cesta z r do všech ostatních vrcholů

Kontext: Min-cost arborescence
Alg: Edmonds' branching algorithm

Reduced weight function

Funkce redukované váhy w' pro directed graf D a funkci váhy w je definována pro každou hranu $e = (u, v) \in E$ jako $w'(e) = w(e) - w(e_v)$

- kde e_v je nejlehčí hrana směřující do v

Kontext: Min-cost arborescence
Alg: Edmonds' branching algorithm

Flow

Flow f je funkce $f : E \to R^+$ uspokojující

- **capacity constraints**
  $\forall e \in E : 0 \leq f(e) \leq c(e)$ kde $c(e)$ je funkce kapacity
- **conservation constraints**
  $\forall v \in V \setminus \{s, t\} : \sum_{(u,v) \in E} f(u, v) = \sum_{(v,w) \in E} f(v, w)$ kde *s* je source, a *t* je sink

hodnota flow f je $|f| = \sum_{(s,v) \in E} f(s, v) = \sum_{(w,t) \in E} f(w, t)$

Kontext: Max. flow
Alg: Dinic, Ford-Fulkerson,...

S-T cut

S-t cut ve flow network *(G, c, s, t)* je birartition vrcholů $S \cup T = V$ taková, že $s \in S$ a $t \in T$.

Kapacita řezu *S,T* je $cap(S, T) = \sum_{(u,v) \in E,\, u \in S, v \in T} c(u, v)$

Kontext: Max flow (= minimum cut)
Alg: ??

Residual graph / network

Residual network pro flow f je dána jako

- **Reziduální graf** $G_f = (V, E_f)$, kde E_f obsahuje pro každou hranu $e = (u, v) \in E$
  - hranu $e = (u, v)$ pokud $f(e) < c(e)$
  - hranu $e^R = (v, u)$ pokud $f(e) > 0$
- **Reziduální kapacita** c_f

$$c_f(e) = \begin{cases} c(e) - f(e) & \text{pokud } e \in E_f \\ f(e) & \text{pokud } e^R \in E_f \end{cases}$$

Kontext: Max flow
Alg: Dinic, Ford-Fulkerson

Augmenting path / bottleneck capacity

Augmenting path je cesta *P* z *s* do *t* v G_f
Bottleneck capacity δ augmentační cesty *P* je minimální kapacita mezi hranami z *P*

Kontext: Max flow
Alg: Dinic,...

Level graph

Level graph $L_G = (V, E_G)$ je podgraf *G* takový, že:

$$E_G = \{(u, v) \in E \mid \exists i \in N . u \in L_i \land v \in L_{i+1}\}$$

kde $L_i = \{v \in V \mid dist(s, v) = i\}$

Kontext: Výpočet blocking flow
Alg: Dinic,..

Blocking flow

Flow f je **blocking** právě když každá $s - t$ cesta v G obsahuje minimálně jednu saturovanou hranu.

Alg: Dinic

Vertex capacity

Dává maximální flow skrze zadaný vrchol

$$c(v) = min(\sum_{(u,v) \in E} c(u,v), \sum_{(v,w) \in E} c(v,w))$$

Kontext: Max flow
Alg: MPM

Matching (maximum/perfect)

Matching M grafu G is podmnožina hran taková, že žádné dvě hrany v M nesdílí vrchol.
Maximum matching je matching o maximální velikosti
Perfect matching je matching který pokrývá všechny vrcholy

Kontext: Matching
Alg: PerfectBiparte,

Unit/Simple capacity network

Network je **unit capacity network (type 1 network)** právě když každá hrana má kapacitu 1
Network je **unit capacity simple network (type 2 network)** pokud:

- každá hrana má kapacitu 1
- každý vrchol $v \in V \setminus \{s,t\}$ má:
    - jednu příchozí hranu
    - jednu odchozí hranu

Kontext: Matching na unit networks
Alg: ??

Gomory hu tree

Pro network $G = (V, E)$ s kapacitami c je **Gomory hu tree strom** $T = (V, F)$ s kapacitami w takovými, že $\forall s, t \in V(G)$:

- $f_G(s,t) = f_T(s,t)$ aka T je ekvivalent flow grafu
- minimální $s - t$ cut v T je zároveň minimální $s - t$ cut v G

Kontext: Min cut
Alg: Gomory hu

Minimum (global) cut

Pro daný connected, undirected graf $G = (V, E)$ a funkce kapacity $c : E \rightarrow R^{+}$, nalezněme množinu $A \subseteq E$ takovou, že $A = \delta(S)$, $\varnothing \subset S \subset V$ a $c(A)$ je minimalizováno.

Kde:

- $c(A) = \sum_{a \in A} c(a)$
- $\delta(X)$ je minimální $v - w$ cut

Kontext: Min cut
Alg: Stoer-Wagner

M-exposed/covered vertice/edge

Matching M **covers** vrchol $v \in V$ pokud nějaká hrana z M je ve vztahu k v (jinak je vrchol exposed)

M-exposed hrana e je taková hrana, která pro M z grafu G $e \in G \wedge e \notin M$

Kontext: Matching
Alg: Edmods bloosom

M-alternating/M-augmenting path

Pro zadaná matching M pro G, cesta P je **M-alternating** pokud hrany střídavě jsou v M a nejsou v M.

Pokud oba konce cesty P jsou rozdílné a M-exposed, řekneme že cesta je **M-augmenting**

M-alternating tree

Nechť M je matching. Řekneme, že rooted tree T (s *root r*) je **M-alternating** pokud T je podgraf G a následující platí:

- každý **vrchol** z T **jiný než *r*** je covered hranou z $M \cap E(T)$
- pro každý vrchol z $v \in V(T)$ je cesta z *r* do *v* M-alternating

Kontext: Matching
Alg: Edmonds bloosom

Frustrated tree

M-alternating tree T je **frustrated** pokud každá hrana má jeden konec v $B(T)$ a druhý v $A(T)$

Vysvětlivky:
- $A(T)$ - vrcholy v liché vzdálenosti od *r*
- $B(T)$ - vrcholy v sudé vzdálenosti or *r*

Kontext: Matching, Max Mathing
Alg: Edmonds bloosom, Edmonds bloosom pro maximum matching

Circular arc colouring depth

Maximální počet cest sdílející jednu hranu

Kontext: Corcular arc colouring

Tree decomposition

Tree decomposition grafu G je dvojice $(T, \{X_t \mid t \in T\})$, kde T je (rooted) tree a X_t, pro každé $t \in T$, je množina vrcholů z G uspokojující následující vlastnosti:

- $\bigcup_{t \in T} X_t = V(G)$
  (aka node coverage)
- pro každou hranu $uv \in E(G)$ existuje node $t \in T$ takové, že $\{u, v\} \subseteq X_t$
  (aka edge coverage)
- pro každý vrchol $v \in V(G)$, je podgraf z T vytvořený z nodů t takových, že $v \in X_t$, je neprázdný a connected
  (aka tree property)

Tree width

Width of a tree decomposition $(T, \{X_t\})$ grafu G je definováno jako

$$width(T, \{X_t\}) = max|X_t| - 1$$

Tree width grafu G, $tw(G)$, je minimální šířka jakéhokoliv tree decomposition z G

Nice tree decomposition

Tree decomposition $(T, \{X_t\})$ je **nice** pokud každý node z $t \in T$ je jeden z následujících typů:

- **Leaf**: žádní potomci, $|X_t| = 1$
- **Introduce**: jeden potomek t', $X_t = X_{t'} \cup \{v\}$ pro nějaké $v \notin X_{t'}$
- **Forget**: jeden potomek t', $X_t = X_{t'} \setminus \{v\}$ pro nějaké $v \in X_{t'}$
- **Join**: dva potomci t_1, t_2, $X_t = X_{t_1} = X_{t_2}$

Isomorphism

Isomorfusmus dvou grafů G a H je bijektivní zobrazení $f : V(G) \rightarrow V(H)$ takové, že:

$$\forall u, v \in V(G) : \{u, v\} \in E(G) \Leftrightarrow \{f(u), f(v)\} \in E(H)$$

Rooted tree

Rooted tree je dvojice (T, r), kde:

- T je strom
- $r \in V(T)$ je vybraný root
- **level** - vzdálenost od root

Rooted tree isomorphism - root grafu G musí být namapovaný na root H

Ordinary trees

Eccentricity vrcholu v v grafu G, značeno jako $ecc(V)$, je vzdálenost z v do nejvzdálenějšího vrcholu od v

Central vertex grafu G je vrchol minimální eccentricity

Centre grafu G je podgraf vytvořen centrálními vrcholy