**BI-SPOL-4 Základní pojmy teorie grafů. Grafové algoritmy: procházení grafu do šířky a do hloubky, určení souvislých komponent, topologické uspořádání, vzdálenosti v grafech, konstrukce minimální kostry a nejkratších cest v ohodnoceném grafu**

BI-AG1

**Neorientovaný graf**

Obsah obrázku text

Popis byl vytvořen automaticky

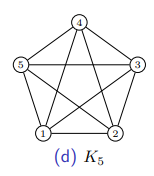
Obsah obrázku text

Popis byl vytvořen automaticky

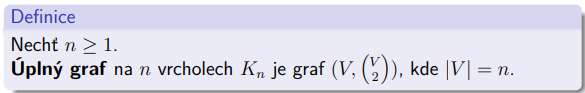
**Orientovaný graf**

**Obsah obrázku text

Popis byl vytvořen automaticky**

**Úplný graf Kn**

* Každý je spojený s každým

****

**Úplný bipartitní graf Kn1,n2**

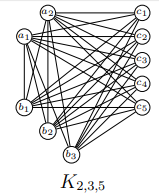
**Obsah obrázku doprava, kolo

Popis byl vytvořen automatickyObsah obrázku text

Popis byl vytvořen automaticky**

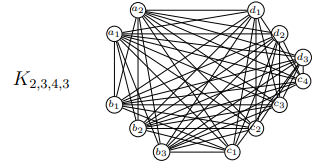
**| E | = n1 \* n2**

**Úplný tripartitní graf**

Obsah obrázku text

Popis byl vytvořen automaticky

**Úplný k-partitní graf**

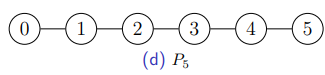
Obsah obrázku text

Popis byl vytvořen automaticky

**Cesta Pm**

* **m je počet hran!**

**Obsah obrázku text

Popis byl vytvořen automaticky**

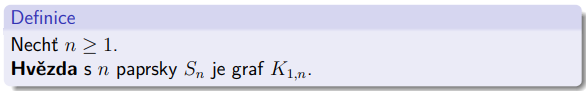
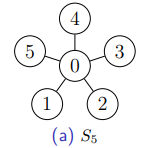
Obsah obrázku šipka

Popis byl vytvořen automaticky**Kružnice Cn**

Obsah obrázku text

Popis byl vytvořen automaticky

**Hvězda Sn**



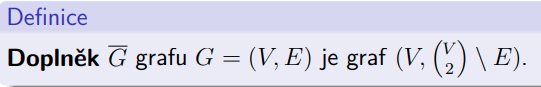
**Značení**

Nechť e = { *u*, *v* } je hrana v grafu G

* vrcholy *u* a *v* jsou koncové vrcholy hrany *e*
* *u* je sousedem *v* v G (a naopak)
* *u* je incidentní s hranou *e* (a *v* je incidentní s *e*)

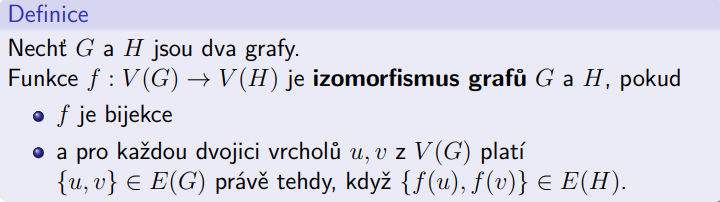
**Doplněk grafu**

* „prohodíme“ hrany



**Izomorfismus**

* Máme dva různě nakreslené grafy – jak poznáme, že jsou stejné? – že lze jeden překreslit, tak že splyne s druhým až na jména uzlů



* **Bijekce** – každému prvku z cílové množiny přiřadí právě jeden prvek ze startovní množiny

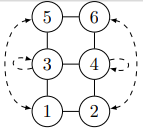
Obsah obrázku text

Popis byl vytvořen automaticky

* Rozhodnout, zdali se jedná o izomorfní grafy je výpočetně náročné (pro nějaké typy grafů ne – cesta)

**Automorfismus**

* graf je izomorfní sám se sebou, ukazují symetrie v grafu
* je to permutace vrcholů neměnící zakreslení grafu

Obsah obrázku text

Popis byl vytvořen automaticky

**Počet grafů**



**Stupeň vrcholu *v*** ( deg(v) ) – počet hran grafu G obsahující vrchol *v*, v orientovaném jsou definované navíc vstupní(+) a výstupní(-) stupně

**Otevřené okolí vrcholu *v*** ( N(v) ) – množina všech sousedů vrcholu *v* (opět vstupní a výstupní)

**Uzavřené okolí vrcholu v** ( N[v] ) – N(v) ∪ {v}

**r-regulární** **graf** – pokud stupeň každého jeho vrcholu je r.

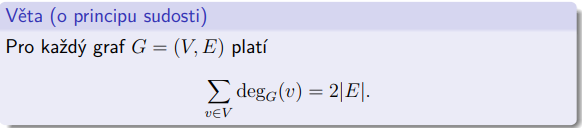
**regulární graf** – pokud je r-regulární pro nějaké r

**izolovaný vrchol** – vrchol stupně 0

**Uzel je zdroj** – vstupní stupeň je 0 (u orientovaných)

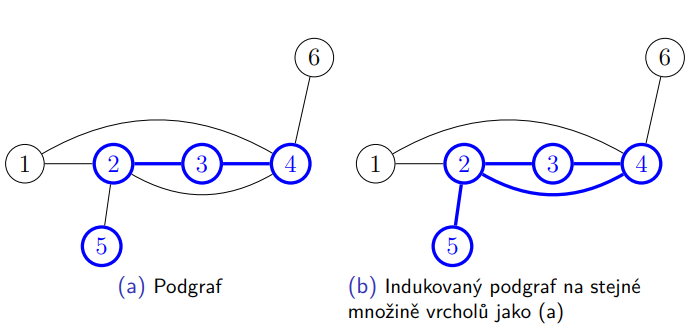
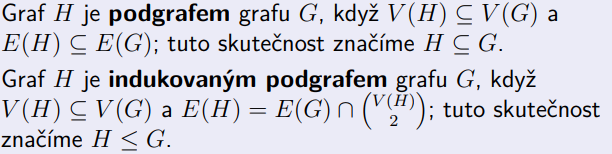
**Uzel je stok** – výstupní stupeň je 0 (u orientovaných)

**Princip sudosti** – suma stupňů všech uzlů je rovna 2\*|E|



**Podgraf** – vybíráme vrcholy a hrany

**Indukovaný podgraf** – vybírá vrcholy a zachovává k nim všechny možné hrany

****

**Klika** – podmnožina vrcholů jež každé dva jsou spojené

**Nezávislá množina** – podmnožina vrcholů z nichž žádné dva nejsou sousední (spojené hranou)

Obsah obrázku text, klipart

Popis byl vytvořen automaticky**Souvislost grafu** – graf G je souvislý, jestliže v něm pro každé jeho dva vrcholy *u, v* existuje *u-v*-cesta.

**Souvislá komponenta** – indukovaný podgraf je souvislou komponentou pokud

* je souvislý a
* neexistuje žádný souvislý podgraf F, F≠ H, grafu G takový, že H ⊆ F

(maximální souvislí podgraf)

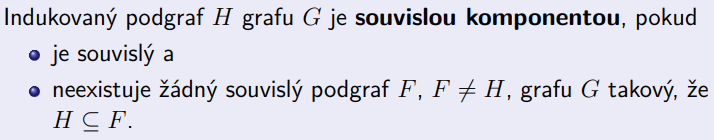
**Vzdálenost**

Obsah obrázku text

Popis byl vytvořen automaticky

Délka cesty = počet hran v cestě

**Souvislá komponenta**



* graf je souvislý právě tehdy, když obsahuje jednu komponentu

**Symetrizace** – nahrazení orientovaných hran neorientovanými

**Slabě souvislý graf** – pokud je symetrizace grafu souvislá

**Silně souvislý graf** – pokud pro každé dva vrcholy *u, v* existuje orientovaná cesta *u → v* a*v → u*

**Strom** – souvislý graf neobsahující kružnici

**List** – vrchol ve stromu se stupněm = 1

**Les** – graf nemá kružnici

**Sled** – posloupnost sousedních vrcholů, ve které se vrcholy mohou opakovat

Obsah obrázku text

Popis byl vytvořen automaticky

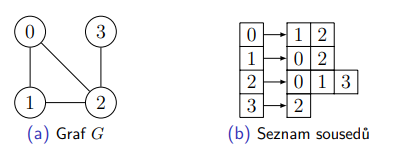
**Délka sledu** – součet délek jeho hran

**Kostra** – nechť G je souvislý graf. Podgraf K grafu G nazveme kostrou grafu G, pokud V(K) = V(G) a K je strom.

**Acyklický (orientovaný)** **graf** je graf neobsahující (orientovanou) kružnici.

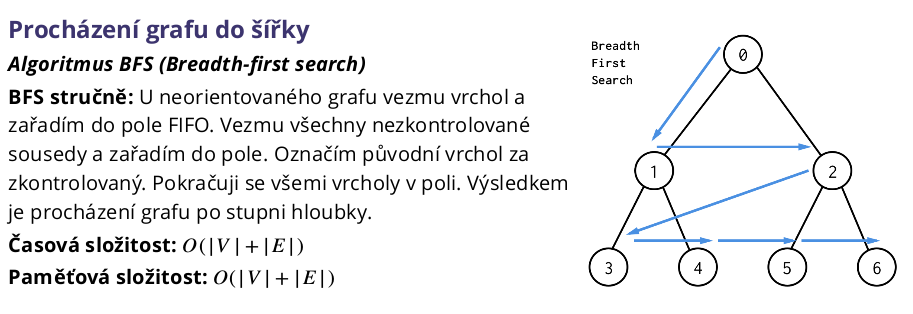
**Reprezentace grafu**

* Matice sousednosti – paměťová s. O()  
  **Obsah obrázku text

  Popis byl vytvořen automaticky**
* Seznam sousedů – paměťová s. O(|V| + |E|)  
  

**BFS**

* Pro konstrukci souvislé komponenty
* Hledání nejkratší cesty (v neohodnoceném grafu)
* Používá frontu (FIFO)

****

**Paměťová složitost**

**Obsah obrázku text

Popis byl vytvořen automaticky**

**Konstrukce nejkratších cest – (hledání souvislé komponenty)**

**Obsah obrázku text

Popis byl vytvořen automaticky**

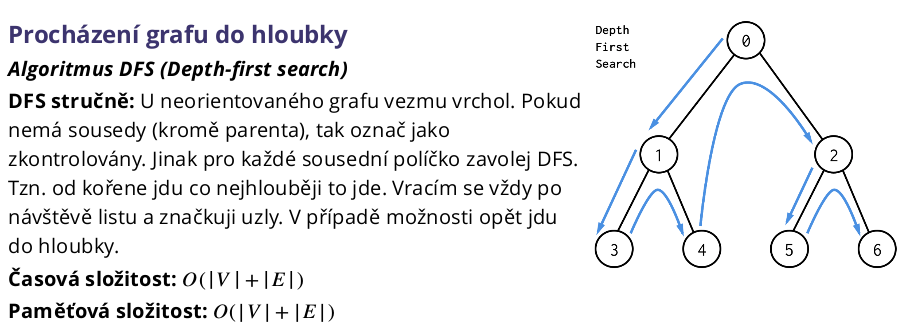
P – pole předchůdců

D – pole vypočtených vzdáleností

Q – fronta vrcholů

**DFS**

* Pro nalezení souvislé komponenty
* Hledání kostry
* Lze napsat rekurzí, nebo iteračně se zásobníkem (LIFO) (nebo i s frontou)

****

**Obsah obrázku text

Popis byl vytvořen automaticky**

**Paměťová složitost**

* reprezentace sousedů zabírá O(|V| + |E|)
* hloubka zanoření rekurze je nejvýše |V|, čili systémový zásobník zabere maximálně O(|V|) paměti
* Celkem tedy O(|V| + |E|)

**Topologické uspořádání**

Obsah obrázku text

Popis byl vytvořen automaticky

* Jinými slovy, pokud vrcholy zakreslíme horizontálně v pořadí topologického uspořádání, budou všechny orientované hrany směřovat zleva doprava.
* Topologické uspořádání se používá především pro plánování pořadí provedení navzájem závislých úloh/výpočtů

Obsah obrázku text, hodinky

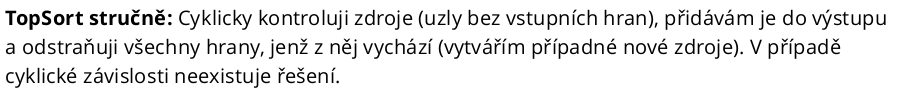
Popis byl vytvořen automaticky

**Orientovaný acyklický graf** – neobsahuje jako podgraf orientovanou kružnici

* Ověřit zdali je graf acyklický můžeme pomocí topSort nebo DFS – ten budeme opakovaně spouštět, než prozkoumáme celý graf (opakované spouštění můžeme nahradit jedním zdrojem – přidáme zdroj, ze kterého povedou hrany do všech vrcholů)
* V grafu existuje cyklus právě tehdy, najde-li DFS alespoň jednu zpětnou hranu

Pokud je G **orientovaný acyklický graf**, potom **obsahuje alespoň jeden zdroj a alespoň jeden stok**

**TopSort**



Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

**Vzdálenosti v grafech**

* V neohodnoceném (orientovaném) grafu je to délka nejkratší (orientované) cesty
  + Najdeme pomocí BFS
* V ohodnoceném grafu je to minimum z délek všech uw-cest
  + Délka = ohodnocení
  + Najdeme pomocí Dijkstra algo. nebo Bellman-fordův algo.

**Obsah obrázku text

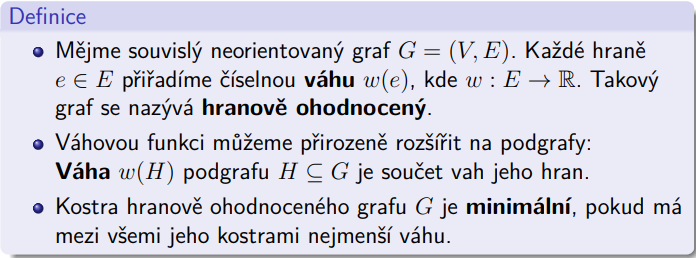
Popis byl vytvořen automaticky**

* neorientovaný graf BFS, orientovaný Dijkstra

**Konstrukce minimální kostry**

* Kruskal a Jarník

**Minimální kostra**

****

**Obsah obrázku text, hodinky, hodiny

Popis byl vytvořen automaticky**

**Jarníkův algoritmus**

* hladový algoritmus – v každém okamžiku vybíráme lokálně nejlepší hranu
* postupně vytváříme strom  
  Obsah obrázku text

  Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

* složitost

Obsah obrázku text

Popis byl vytvořen automaticky

* pro lepší složitost můžeme k implementaci využít binární haldu – opakovaně vybíráme minimum

**Obsah obrázku text, osoba, snímek obrazovky

Popis byl vytvořen automaticky**

* při použití binární haldy je složitost:



**Kruskalův algoritmus**

* hladový algoritmus
* **stručně**: vybírám nejlehčí hrany grafu a přidávám je do kostry, pokud nevytváří cyklus
* začneme lesem tvořeným samotnými vrcholy bez hran a zkoušíme přidávat hrany od nejlehčí po nejtěžší a zahazujeme ty, které by vytvořily cyklus
* po projetí všech hran končí
* nejprve musíme seřadit hrany

Obsah obrázku text

Popis byl vytvořen automaticky

* pro implementaci se používá datová struktura Union-Find
  + tu můžeme implementovat pomocí pole nebo keříků

Obsah obrázku text

Popis byl vytvořen automaticky

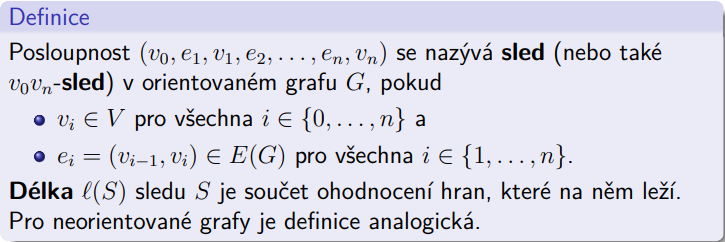
* časová složitost: (m log n + n^2 ), pokud má keříkovou strukturu, tak v O(m log n)  
  Obsah obrázku text

  Popis byl vytvořen automaticky

**Nejkratší cesta v grafu**

* Dijkstra a Bellman-Ford
* NP-úplná úloha – pro obecný graf neznáme efektivní algoritmus
* Lze řešit i s BFS, pokud každou hranu s ohodnocením převedeme na více uzlů a hran s ohodnocením 1, ale tím naroste složitost

**Sled**



Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku šipka

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

**Dijkstrův algoritmus**

<https://www.youtube.com/watch?v=pVfj6mxhdMw&t=100s&ab_channel=ComputerScience>

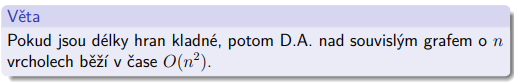
<https://www.youtube.com/watch?v=XB4MIexjvY0&ab_channel=AbdulBari>

* **předpokládá kladné hrany** (se zápornými nemusí fungovat)
* U všech vrcholů označím nejkratší vzdálenost +*inf* z neznámého předchůdce
* vyberu počáteční vrchol *u* a všem následovníkům zapíšu délku cesty a předchůdce označím jako dokončený
* vezmu vrchol z nedokončených s **nejmenší délkou** cesty a pro jeho následovníky spočítám délku tak, že vezmeme aktuální délku (vrcholu) + délka hrany. Pokud bude menší, než aktuálně má následovník, tak nahradíme a označíme následovníka jako otevřeného (abychom zaktualizovali cestu i pro jeho následovníky). Aktuální vrchol uzavřeme – **relaxace vrcholů**
* pokračuji, než uzavřu všechny vrcholy

Obsah obrázku text

Popis byl vytvořen automaticky

* složitost - |V| \* |V| - procházíme všechny (n) vrcholy – pro každý se můžeme zeptat n vrcholů (sousedů)



Brzdí nás hledání minima

* implementovat můžeme i binární haldou – lepší časová složitost

Obsah obrázku text

Popis byl vytvořen automaticky

**Bellman-Ford**

* předpokládá neexistenci záporných cyklů (ale připouští záporné délky)
* princip jako Dijkstra, ale nepoužívá se prioritní fronta pro výběr uzlů s nejkratší délkou, ale klasická fronta
* Jinými slovy, otevřený vrchol ke zpracování (relaxaci následníků) zvolíme vždy jako nejstarší (tedy první) vrchol ve frontě otevřených vrcholObsah obrázku text

  Popis byl vytvořen automaticky
* Postup viz. papír
* Obsah obrázku text

  Popis byl vytvořen automaticky
* Obsah obrázku text

  Popis byl vytvořen automaticky