

# Algoritmy a Dátové Štruktúry

Jana Katreniaková

`katreniakova@dcs.fmph.uniba.sk`

## Graf

Graf je dvojica  $(V, E)$ , kde

- $V$  je množina vrcholov
- $E$  je množina hrán – dvojíc navzájom prepojených vrcholov

## Hrany

Hrany v grafe môžu byť

- usporiadané dvojice vrcholov  $e = (u, v) \in V \times V$  (orientovaný graf)
- neusporiadané dvojice vrcholov  $e = (u, v)$  (neorientovaný graf)

## Graf

Graf je dvojica  $(V, E)$ , kde

- $V$  je množina vrcholov
- $E$  je množina hrán – dvojíc navzájom prepojených vrcholov

## Hrany

Hrany v grafe môžu byť

- usporiadané dvojice vrcholov  $e = (u, v) \in V \times V$  (orientovaný graf)
- neusporiadané dvojice vrcholov  $e = (u, v)$  (neorientovaný graf)
  - musí platiť  $(u, v) \in E \Rightarrow (v, u) \in E$

- $e = (u, v) \in E$  potom vrcholy  $u, v$  sú incidentné a sú incidentné s hranou  $e$  (a naopak)
- stupeň vrchola počet incidentných hrán (pri orientovanom grafe vstupný resp. výstupný stupeň)
- viacnásobné hrany – ak existujú  $e_1 \neq e_2$  medzi tými istými vrcholmi
- loop
- cesta/cyklus
- ..

## Možnosti

- Zoznam hrán
- Zoznam hrán pre jednotlivé vrcholy
- Matica susednosti (binárna resp. s hodnotami hrán)
- Komplikovanejšie štruktúry

## Operácie, ktoré nás zaujímajú

- `incidentEdges(v), incidentVertices(v)`
- `areAdjacent(u,v)`
- prípadne `insert/remove edge/vertex`

## Možnosti

- Zoznam hrán
- Zoznam hrán pre jednotlivé vrcholy
- Matica susednosti (binárna resp. s hodnotami hrán)
- Komplikovanejšie štruktúry

## Operácie, ktoré nás zaujímajú

- `incidentEdges(v)`, `incidentVertices(v)`
- `areAdjacent(u,v)`
- prípadne `insert/remove edge/vertex`

- Prechádzanie grafu po hranách medzi vrcholmi
- Dostávame 'dostupné' vrcholy v určitom poradí
- Hrany, ktoré objavili vrchol sú istým spôsobom významné (koľko ich je?)

Skúma nejakú časť až k jej koncu. Základnou vlastnosťou tohto algoritmu je snaha o maximálne preskúmanie zvolenej cesty pred prípadnou voľbou novej cesty.

$DFS(G = (V, E), v_0)$

- 1  $Visited \leftarrow Visited + \{v_0\}$
- 2 *spracuj*( $v_0$ )
- 3 **for**  $w \in incidentVertices(v_0)$
- 4 **do if**  $w \notin Visited$
- 5       **then**  $DFS(w)$

Dá sa simulovať zásobníkom.



Vrcholy prechádzame v poradí závislom od vzdialenosti (počet hrán) od počiatku (t.j. najprv susedov, potom ich susedov).  
Dosiahneme to keď zásobník nahradíme frontou  $Q$ .

BFS( $G = (V, E), v_0$ )

```
1  enqueue( $Q, v_0$ )
2  while not(isEmpty( $Q$ ))
3  do  $v \leftarrow$  dequeue( $Q$ )
4     if  $v \in$  Visited
5     then continue;
6     spracuj( $v$ ); Visited  $\leftarrow$  Visited + { $v$ };
7     for  $w \in$  incidentVertices( $v$ )
8     do if  $w \notin$  Visited
9         then enqueue( $Q, w$ );
10
```



Ako s prehľadávaniaми

Topologické triedenie

Ako s prehľadávaniaми

Topologické triedenie