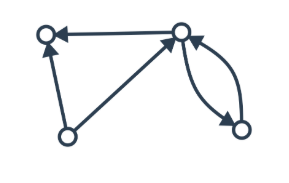
# Definice grafu

## Orientovaný

V orientovaném grafu mají hrany přesně určený směr, kterým vedou, a tedy i začáteční a koncový vrchol.



## Neorientovaný

V neorientovaném grafu nemají hrany určený směr, kterým vedou.

Obsah obrázku řada/pruh, diagram, kruh, design

Obsah generovaný pomocí AI může být nesprávný.

## Ohodnocený

V ohodnoceném grafu má každá hrana přiřazenu určitou hodnotu (nazývanou také váha).

**Obsah obrázku řada/pruh, diagram

Obsah generovaný pomocí AI může být nesprávný.**

# Kostra grafu

Kostrou grafu budeme rozumět libovolný [podgraf](https://teorie-grafu.cz/zakladni-pojmy/podgraf.php), který hranami spojuje **všechny vrcholy** původního grafu a zároveň sám neobsahuje žádnou [kružnici](https://teorie-grafu.cz/zakladni-pojmy/kruznice-cyklus.php) (→ jde o [strom](https://teorie-grafu.cz/zakladni-pojmy/stromy.php)).

Obsah obrázku řada/pruh, diagram, Symetrie, origami

Obsah generovaný pomocí AI může být nesprávný.

# Matice sousednosti vs seznam sousedů

## Matice sousednosti

Je to **čtvercová tabulka** (matice), kde řádky a sloupce odpovídají vrcholům grafu.

**Matice** je jako tabulka „kdo je s kým propojen“.

Obsah obrázku diagram, Plán, řada/pruh

Obsah generovaný pomocí AI může být nesprávný.

## Seznam sousedů

Každý vrchol má svůj **seznam vrcholů**, ke kterým je připojen hranou.

**Seznam** je jako „u každého vrcholu si piš, s kým se baví“.

Obsah obrázku diagram, účtenka

Obsah generovaný pomocí AI může být nesprávný.

# Reálné využití teorie grafů

**Města/křižovatky jsou vrcholy, silnice/trasy jsou hrany.**

Grafové algoritmy pomáhají najít:

* nejkratší cestu
* nejrychlejší trasu (s ohledem na dopravu)
* objížďky nebo alternativy

**Rozvrhy, výrobní procesy, závislosti mezi úkoly**

* kdy je možné co začít (graf závislostí)
* jak minimalizovat dobu výroby nebo provádění

**Mnoho her (např. sudoku, šachy, logické úlohy) se dá modelovat pomocí grafů.**

* hledání tahů
* ověřování pravidel
* vývoji herní AI

# Problém hledání nejkratší cesty

## Co znamená „nejkratší cesta“

Pokud jsou hrany **neohodnocené** (neobsahují váhy), pak „nejkratší“ = **nejméně hran** mezi dvěma vrcholy.

Pokud jsou hrany **ohodnocené** (např. vzdálenost, čas, cena), pak „nejkratší“ = **nejmenší součet vah**.

## Negativní hrany a jejich vliv

Hrany s **negativní vahou** znamenají, že zisk místo ztráty.

* Ekonomika: transakce s výdělkem/ztrátou.
* Biologie: reakce, kde se něco ušetří (energeticky výhodné).
* Herní systém: „získat bonus“ za konkrétní tah.

# Přehled algoritmů

## Dijkstrův algoritmus

* Princip (greedy přístup, prioritní fronta)
* Omezení (nefunguje s negativními hranami)
* Časová složitost: **O(E + V log V)**, kde E je počet hran a V je počet vrcholů v grafu

## Bellman-Fordův algoritmus

* Princip (relaxace hran, až V−1 iterací)
* Výhoda: funguje i s negativními hranami
* Detekce záporných cyklů
* Časová složitost: **O(V \* E)**, kde V je počet vrcholů a E je počet hran v grafu

## Floyd-Warshallův algoritmus

* Princip (dynamické programování)
* Vhodný pro všechny páry vrcholů
* Časová složitost: **O(n³)**, kde n je počet vrcholů v grafu

## Implementace (praktická část)

<https://github.com/Gabi1818/Grafove-algoritmy-pro-hledani-nejkratsi-cesty>

Příklad grafu pro testování:

Obsah obrázku řada/pruh, diagram, kruh, design

Obsah generovaný pomocí AI může být nesprávný.

Zdroje:

[Teorie grafů](https://teorie-grafu.cz/)

[Problém sedmi mostů města Královce](https://www.algoritmy.net/article/91/Problem-sedmi-mostu)