

# Dokumentace

- IAL\_kostra\_grafu
- IAL, Náhradní projekt - 07. Minimální kostra grafu
- Kulinkovich Andrei, Marochkina Elena, Tréšek Roman
- src -hlavní adresář
  - implementace algoritmu rozdělena do několika jednotlivých souborů, pro jednotlivé části grafu
    - graphs - adresář s testovými soubory
- datum vytvoření: 12.11.2023, datum poslední změny: 19.11.2023

## Zadání

07 . Minimální kostra grafu - Vytvořte program pro hledání kostry grafu s minimálním ohodnocením pro ohodnocené neorientované grafy.

## Řešení

K řešení jsme využili Primova algoritmu, jeho teoretická složitost je  $O(E + V \cdot \log_2 E)$ , kde  $E$  je počet hran a  $V$  je počet vrcholů v grafu.

## Postup algoritmu

Ze začátku se náhodně vybere jeden uzel jako počáteční, zbylé uzly se zapíší do pole nezpracovaných uzlů. Všechny hrany, které vedou z počátečního uzlu se zapíší na seznam dostupných hran. Algoritmus projde seznam dostupných hran a vybere tu s nejnižším ohodnocením, která ale musí vést do ještě nezpracovaného uzlu. Hrana je poté odstraněna ze seznamu dostupných hran a uzel se odstraní z pole nezpracovaných uzlů. Zpracovávaná hrana se přidává do fronty hran reprezentujících postupně rostoucí kostru grafu. Uzel do kterého zpracovávaná hrana vedla se odstraní z pole nezpracovaných uzlů. Všechny hrany dostupné z nového uzlu jsou zapsány na seznam dostupných hran. Algoritmus probíhá tak dlouho, dokud nezpracuje všechny uzly.

## Složitost implementovaného algoritmu

Složitost námi implementovaného algoritmu je  $O(V^2 + E)$ , protože jsme využili implementaci pomocí seznamu sousedů oproti rychlejší variantě s využitím haldy.

Implementace se seznamem sousedů je pomalejší, protože prochází seznam uzlů a hledá jeho minimum, což může v nejhorším případě mít složitost až  $O(V^2)$ .