# Kruskalův Algroritmus

Jakub Kacálek

2022

4. 4. 2022

# Obsah

Úvod 3.

Užití 4.

Složitost 4.

Princip 4.

Kód 5.

Zdroje 6.

# Úvod

Kruskalův algoritmus využívaný v teorii grafů je využívaný k nalezení minimální kostry grafu s nezápornými hranami. Kruskalův algoritmus je příkladem hladového algoritmu.

## Užití

Užití Kruskalova algoritmu můžeme nalézt například v elektrických sítích a nebo na lokálních sítích .

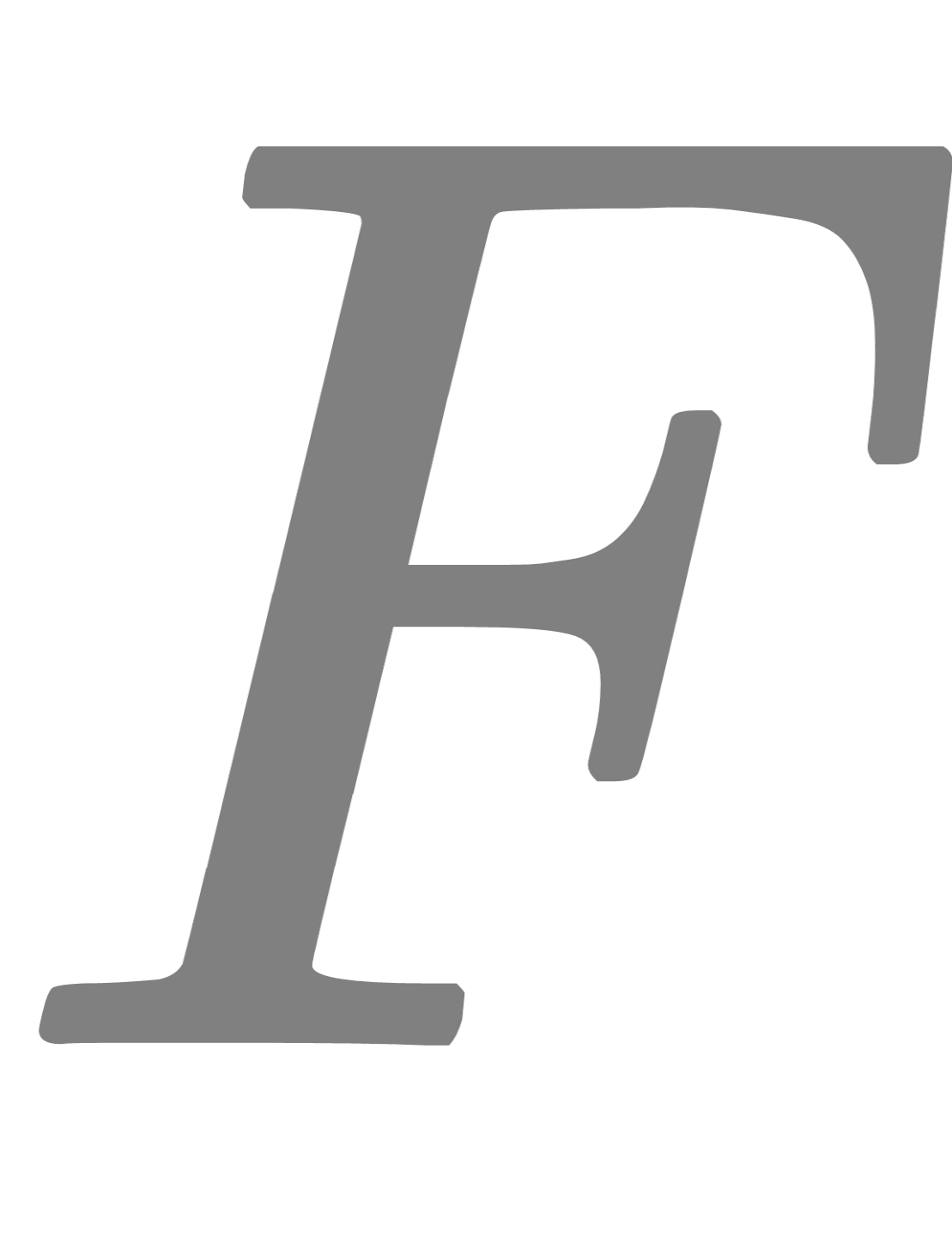
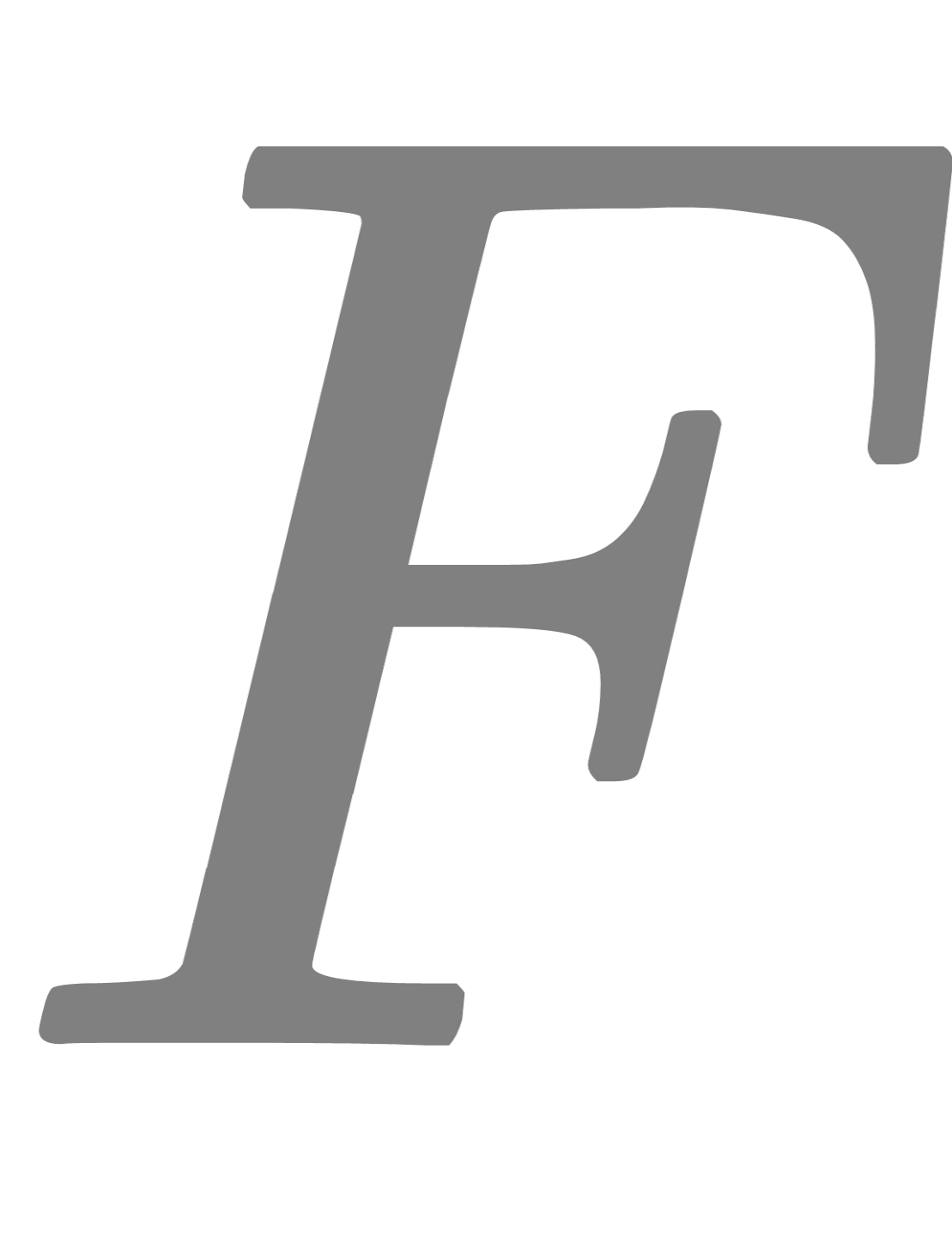
**Složitost**

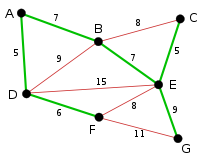
Časová náročnost závisí na způsobu implementace sorting algoritmu

Samostatný algoritmus má O ( H log H), kde H je hrana.

**Princip**

Algoritmus můžeme popsat tak, že vždy vybírá taková hrana, která má ze všech možných hran spojujících dva různé postromy ve vytvářené kostře tu nejmenší váhu.

* vytvoř les  (množinu stromů), ve kterém je každý uzel grafu samostatným podstromem
* vytvoř množinu  obsahující všechny hrany grafu
* dokud je množina  neprázdná
* z množiny  odeber hranu s minimální váhou
* pokud tato hrana spojuje dva různé podstromy, přidej ji do lesa , tak že tyto podstromy sluč do jednoho
* v jiném případě
* hranu zahoď



**Kód**

1 Kruskal(G,w)

2 A := ∅ Vybraná kostra zatím prázdná,

3 **for** každý uzel u ∈ U //pro každý uzel se vytvoří samostatný podstrom

4 **do** MAKE-SET(u)

5 uspořádej H do neklesající posloupnosti podle váhy w

6 **for** každou hranu [u, v] ∈ H v pořadí neklesajících vah

7 **do** **if** FIND-SET(u) != FIND-SET(v) //Hrana [u, v] je vhodná, přidej ji do kostry a spoj odpovídající podstromy

8 **then** A := A ∪ {[u, v]}

9 UNION(u, v)

10 **return** A

Implementované řešení může být nalezeno na <https://github.com/Kaciis/kruskal>

**Zdroje**

<https://en.wikipedia.org/wiki/Joseph_Kruskal>

<https://cs.wikipedia.org/wiki/Kruskal%C5%AFv_algoritmus>

<https://en.wikipedia.org/wiki/Kruskal%27s_algorithm>

<https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/>

<https://www.algoritmy.net/article/1417/Kruskaluv-algoritmus>