

projekt do předmětu GAL – Grafové algoritmy

# Dinicův algoritmus

## Demonstrační úloha

řešitelé: **Petr Svoboda**, xsvobo91  
**Martin Matoušek**, xmatou21

### Použité zdroje a knihovny

Demonstrační aplikace byla napsána v jazyce C++ a pro vytvoření grafického rozhraní byla použita knihovna [Qt](#). Pro ukládání, načítání a výpočet rozložení grafu byla použita knihovna [OGDF](#).

Informace o algoritmu, jeho vlastnostech a jeho modifikacích jsme čerpali z [osobních stránek](#) tvůrce algoritmu Yefima Dinitze.

Při tvorbě aplikace samotné jsme vycházeli a inspirovali se z ilustračních příkladů knihovny Qt, pro samotný algoritmus jsme použili [tento zdroj](#), který jsme pro naše účely upravili.

### Minimální požadavky

Vzhledem k použití knihovny Qt je aplikace multiplatformní a dále nezávislá na jiných knihovnách. K překladu zdrojových souborů byl použit překladač [gcc](#). Projekt samotný byl testován na operačním systému Linux.

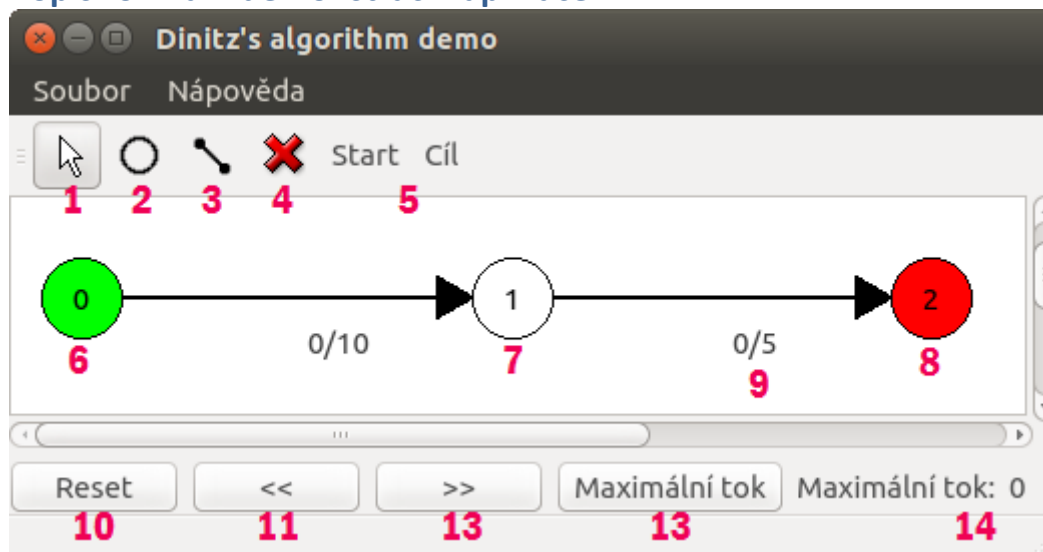
### Úvod k Dinicovu algoritmu

Je vhodné uvést několik málo poznámek k samotnému Dinicovu algoritmu. Veškeré následující informace jsou čerpány z publikace „[Dinitz' Algorithm: The Original Version and Even's Version](#)“ samotného autora algoritmu Yefima Dinitze z jeho osobních stránek zmíněných v úvodu.

Samotný algoritmus vychází z Ford-Fulkersonova algoritmu a je jeho modifikací. Na konci sekce 2.2 ve výše zmíněném dokumentu je popis originálního, původního Dinicova algoritmu. Tento původní algoritmus byl však během let modifikován dalšími lidmi, přičemž modifikace se týkaly především konkrétních implementačních kroků. Jedna z nejvíce populárních modifikací je od pánů Shimona Evena a Alona Itai, která v implementaci využila již známé algoritmy BFS a DFS.

Nad co nejefektivnější implementací se zamýšlel také Boris Cherkassky, jehož úprava, resp. jistá modifikace, je použita v projektu. Je nutno podotknout, že tento algoritmus využívá opět již zmíněné algoritmy DFS a BFS.

## Popis rozhraní demonstrační aplikace



Obrázek 1 - Rozhraní

Na Obrázku 1 je okno hlavního programu s popisem jednotlivých grafických prvků, se kterými lze interagovat. Jejich popis je následující:

1 – při aktivaci tohoto tlačítka můžete kliknutím a tahem posouvat jednotlivé uzly (na obr. popisky 6,7 a 8), dvojklikem na 9 měnit kapacitu hrany a případně klikem na uzly je vybírat. Kolem vybraného uzlu se vykreslí čtverec, který označuje vybraný uzel. Tento vybraný uzel pak lze označit pomocí tlačítek 5 „Start“ a „Cíl“ jako zdroj a spotřebič. Vybírat lze také hrany, které se označí stejným způsobem jako uzly.

2 – kreslení uzlů. Při aktivaci tohoto tlačítka a následném kliknutí na plátno se na daném místě vykreslí uzel.

3 – kreslení hran. Při aktivaci tohoto tlačítka a kliknutí a tahu od jednoho uzlu k druhému se mezi těmito uzly vykreslí hrana s daným směrem kreslení.

4 – tlačítko pro mazání hran či uzlů. Kliknutím křížku na jednotlivé uzly či hrany se tento objekt vymaže.

5 – tlačítka pro označování zdroje (Start) a spotřebiče (Cíl). Při označování musí být daný uzel vybraný (viz 1)

6 – zelený uzel značí zdroj

7 – bílý, nezabarvený uzel značí „obyčejný“ uzel

8 – červený uzel značí uzel spotřebitele

9 – hrana s hodnotami ve formátu „tok/kapacita“. Kliknutím na daný popis (číslo) lze editovat kapacitu pomocí vstupu z klávesnice. Zadává se v tomto případě pouze jediné číslo, bez dalšího formátování, které označuje kapacitu.

10 – restart všech kroků, uvádí graf do počáteční konfigurace.

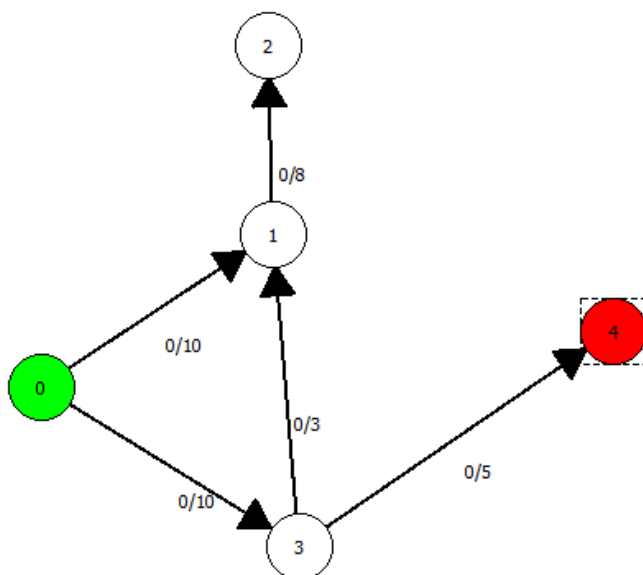
11 – krok zpět

13 – krok vpřed „>>“, který lze krokovat zpětně pomocí 11 a tlačítko „Maximální tok“ což výsledné řešení, které nelze krokovat zpět

### Vysvětlení grafického řešení úlohy

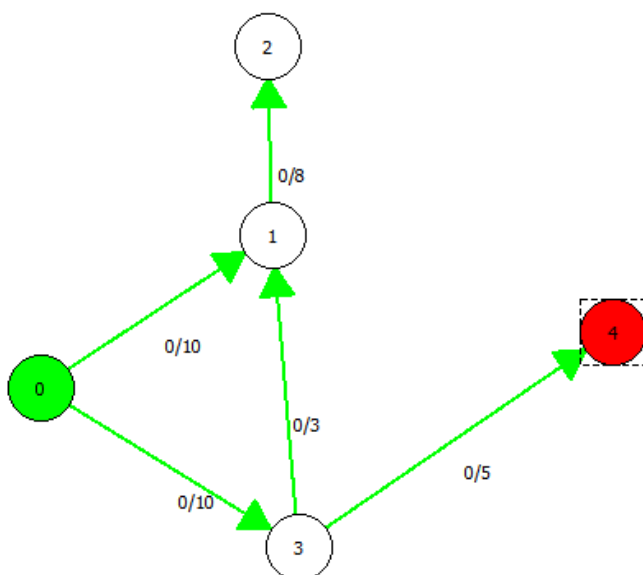
Při hledání maximálního toku jednotlivých úloh nad grafem se mění zbarvení hran jednotlivých hran při krokování algoritmu a navíc zbarvení výsledného řešení u postupného krokování se liší od výsledného zbarvení při zmáčknutí tlačítka 13, tedy kompletního řešení.

#### Postupné krokování



Obrázek 2

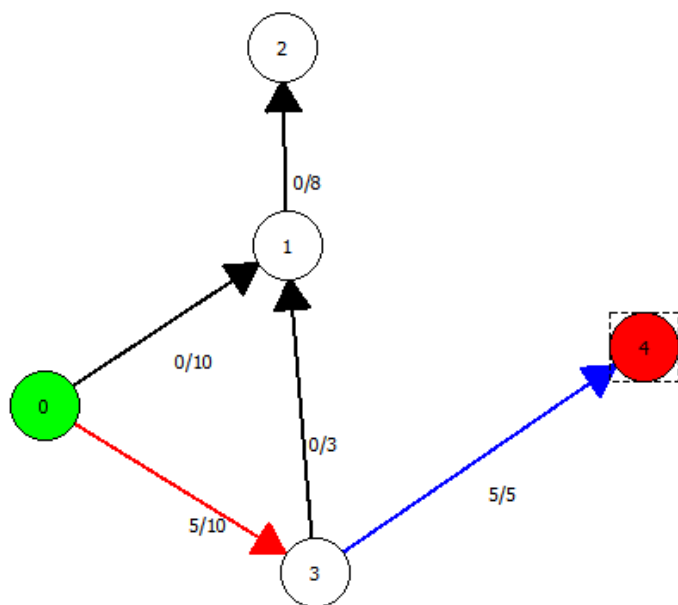
Na obrázku 2 vidíme počáteční stav úlohy s hranami nastavenými na své kapacity a označenými uzly zdroj a spotřebič.



Obrázek 3

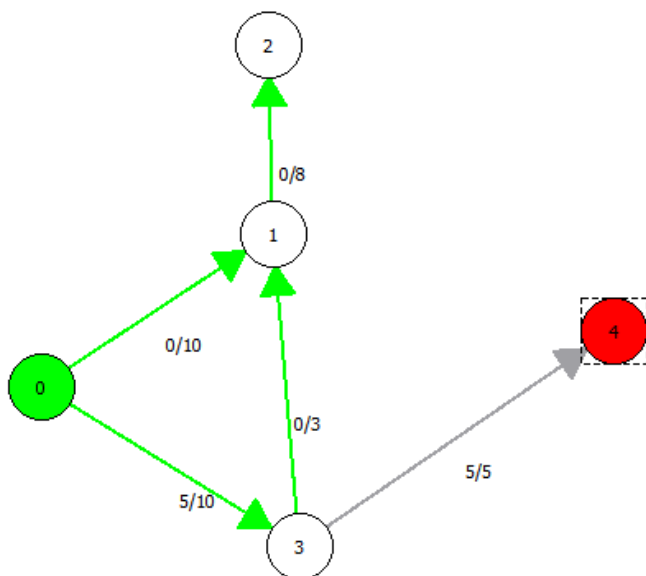
Obrázek 3 ilustruje graf po první kroku algoritmu. Zde se zelenými šipkami vykreslí všechny možné cesty, kterými se lze v grafu pohybovat. Tato ilustrace vždy neodpovídá výstupu algoritmu BFS. Jde pouze o ilustraci, zda se dá dostat v daném kroku do koncového

uzlu. Tato ilustrace bude užitečnější a názorněji bude ilustrovat princip fungování algoritmu s přibývajícími kroky.



Obrázek 4

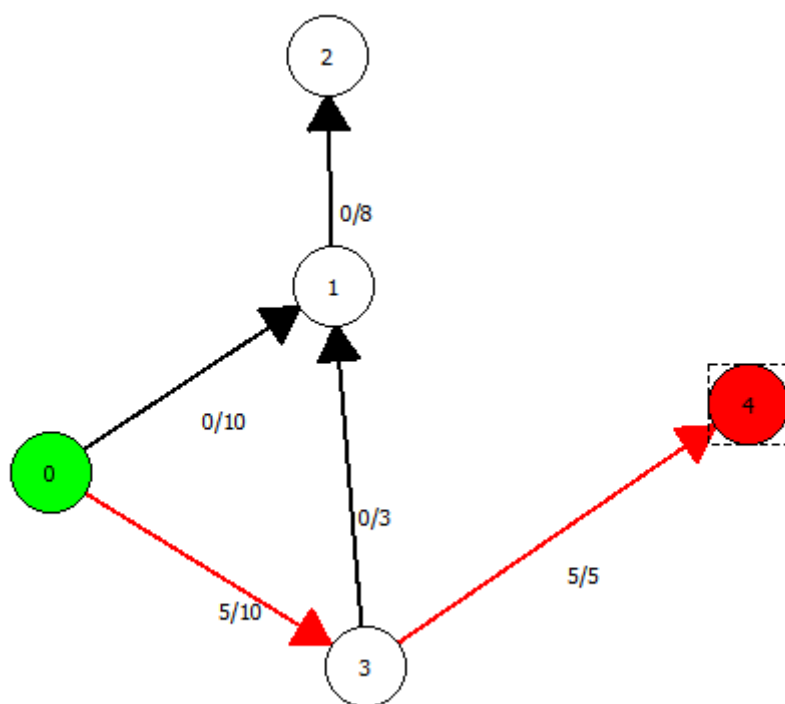
Na tomto obrázku vidíme, že původní zelené barvy se změnily v obyčejné černé a cesta ze zdroje do spotřebiče má nyní dvě barvy, červenou a modrou. Červená značí samotnou barvu cesty od zdroje ke spotřebiči a modrá barva na této cestě znázorňuje hranu, na které se naplnila kapacita a jejíž hodnota je minimem na dané cestě.



Obrázek 5

Finální obrázek řešení. Šedé hrany zdůrazňují maximální využitou kapacitu hrany a zelené ukazují, že ze zdroje se již nedá dostat do spotřebitele a tedy algoritmus tímto končí.

## Kompletní vyřešení úlohy



Obrázek 6

Na obrázku 6 je vyobrazeno konečné řešení při stisku tlačítka 13. Jsou zde pouze červené a černé hrany. Červené hrany znázorňují cestu, po které „je distribuován“ vypočítaný maximální tok.

## Překlad a spuštění

V archivu se zdrojovými soubory je přiložen soubor README.txt, který obsahuje postup pro překlad projektu v závislosti na platformě. Projekt byl vyzkoušen na školním serveru merlin.