

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

## FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Projekt IAL, 2018Z

## **Obarvení grafu**

Projekt č.6

30. listopadu 2018

**Tým:**

Adámek Josef, xadame42

Barnová Diana, xbarno00

Vanický Jozef, xvanic09

Weigel Filip, xweige01

# Obsah

<b>1</b>	<b>Zadanie</b>	<b>2</b>
<b>2</b>	<b>Práca v tíme</b>	<b>2</b>
2.1	Príprava a plán . . . . .	2
2.2	Postup a rozdelenie práce . . . . .	2
2.3	Komunikácia . . . . .	2
<b>3</b>	<b>Teoretická časť</b>	<b>2</b>
3.1	Priblíženie problematiky . . . . .	2
3.2	Popis projektu . . . . .	3
<b>4</b>	<b>Testování projektu</b>	<b>3</b>
<b>5</b>	<b>Implementácia</b>	<b>3</b>
<b>6</b>	<b>Záver</b>	<b>3</b>
<b>7</b>	<b>Zdroje</b>	<b>3</b>

# 1 Zadanie

Vytvoriť program pre hľadanie **minimálneho zafarbenia neorientovaných grafov**. Ak existuje viacej riešení, stačí nájsť len jedno. Výsledky prezentujte vhodným spôsobom. Súčasťou projektu bude načítavanie grafov zo súboru a vhodné testovacie grafy. V dokumentácii uveďte teoretickú zložitosť úlohy a porovnejte ju s experimentálnymi výsledkami.

## 2 Práca v tíme

### 2.1 Príprava a plán

Pred začiatkom vývoja boli vziať do úvahy schopnosti každého člena tímu, na základe ktorých boli pridelené úlohy. Časové rámce jednotlivých častí boli len orientačné pre udržanie prehľadu nad postupom a zostávajúcim časom do ukončenia projektu. Byl vytvorený privátny repozitár na Githube, fungujúci na technológii Git pre ľahké verzovanie projektu. Možnosti vytvoriť jednotlivé vetvy pro každú súčasť a nezávislý vývoj a pre prípadnú orientáciu medzi verziami či vrátenia k predchádzajúcej verzii. Pro statickú analýzu kódu sme využili službu Codacy.

### 2.2 Postup a rozdelenie práce

Pri tomto projekte sme sa rozhodli využiť metódu Test-driven development a to z dôvodu zefektívnenia celého vývoja. Najskôr bol napísaný test k súčasti, ktorá bola až následne naprogramovaná. Tak sa zabránilo k zavedeniu chyby do už funkčnej časti.

### 2.3 Komunikácia

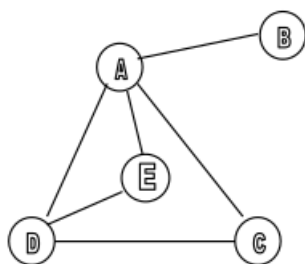
Komunikácia v tíme prebiehala prostredníctvom služby Messenger ale aj osobne na pravidelných stretnutiach, ktoré sme si dohodli už na začiatku riešenia projektu.

## 3 Teoretická časť

### 3.1 Priblíženie problematiky

**Graf** (všeobecne) je definovaný trojicou  $G=(N, E, I)$ , kde

- $N$  je množina uzlov, ktorým je možné priradiť hodnotu
- $E$  je množina hrán, ktorým je možné priradiť hodnotu. Každá hrana spojuje dva uzly a môže byť orientovaná. Ak je hrana orientovaná tak sa hovorí o orientovanom grafe. V našom prípade sa teda jedná o **neorientovaný graf**, pretože hrany sú neorientované.
- $I$  je množina spojenia, ktorá jednoznačne určuje dvojice uzlov daného grafu



	A	B	C	D	E
A	0	1	1	1	1
B	1	0	0	0	0
C	1	0	0	1	0
D	1	0	1	0	1
E	1	0	0	1	0

Obrázek 1: Príklad neorientovaného grafu a jeho matice

**Implementácia** neorientovaného grafu je pomocou matice koincidencie t.j. spojenia. (Táto matica je symetrická podľa hlavnej diagonály.)

**Zafarbením grafu** rozumieme priradenie farieb uzlom grafu, pričom žiadne dve susedné uzly nesmú byť zafarbené rovnako. Minimálny počet použitých farieb sa nazýva **chromatické číslo**. Práve toto chromatické číslo v našom projekte hľadáme.

### 3.2 Popis projektu

**Spúšťanie:** ./main -f FILENAME [-h] [-b], kde

- -f FILENAME je názov súboru, ktorý obsahuje maticu grafu
- -b je voliteľný parameter, ktorý má na starosti...
- -h je voliteľný parameter, ktorý má na starosti vytlačíť spravu pre používanie na stdout

## 4 Testování projektu

Projekt sme najskôr testovali na menších testovacích grafoch, ktoré sme vytvorili a vedeli aj odkontrolovať. Neskôr sme na generovanie grafov vytvorili skript, ktorý vytvoril aj väčšie grafy. Testovanie prebiehalo na platforme Ubuntu, MacOS a FreeBSD.

## 5 Implementácia

TBD

## 6 Záver

TBD

## 7 Zdroje

opora ial, izu prednasky, prednasky ial,