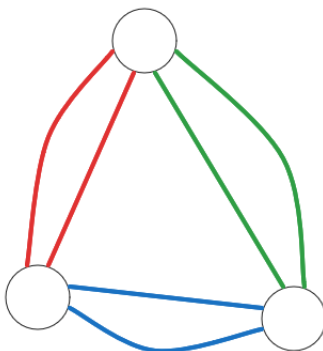


Report ročníkový projekt zimný semester 2024/2025

Alex Diko
diko3@uniba.sk

Pojmy: *Graf* je dvojica $G = (V, E)$, kde V, E sú množiny také, že $E \subseteq V \times V$ a $V \cap E = \emptyset$. Prvky V sa nazývajú *vrcholy* grafu G a prvky E jeho *hrany*. Množinu hrán grafu G označujeme ako $E(G)$. Vrchol v je *incidentný* s hranou e , ak $v \in e$. *Stupeň vrcholu* je počet hrán s ním incidentných. Dve hrany sú *susedné*, ak existuje vrchol, ktorý je s obidvomi incidentný. Ak všetky vrcholy grafu majú stupeň k , graf sa nazýva *k-regulárny*. *Cesta* je neprázdny graf $P = (V, E)$, kde $V = \{x_0, x_1, \dots, x_k\}$ a $E = \{x_0x_1, x_1x_2, \dots, x_{k-1}x_k\}$, pričom všetky x_i sú rôzne. Túto cestu označíme $x_0x_1 \dots x_k$. Ak $P = x_0 \dots x_{k-1}$ je cesta, potom graf $C := P + x_{k-1}x_0$ sa nazýva *kružnica*. *Dĺžka* kružnice je počet jej hrán. Graf $G'(V', E')$ je *podgrafom* grafu G ak $V' \subseteq V$ a $E' \subseteq E$. *Rozklad* množiny $E(G)$ je množina $R = \{R_1, \dots, R_k\}$, kde R_i sú navzájom disjunktné a zjednotenie $\cup R$ všetkých množín $R_i \in R$ sa rovná $E(G)$. [1]

Kružnicovou dekompozíciou grafu nazveme rozklad $E(G)$ na kružnice, ktoré nezdieľajú žiadnu hranu. Ak každá kružnica tohto rozkladu má párnú dĺžku, nazveme tento rozklad *párny* (ďalej *ECD* z anglického *Even cycle decomposition*). Pre ECD grafu zafarbíme každú jeho kružnicu tak, aby kružnice so spoločným vrcholom nemali rovnakú farbu. Potom zjednotenie kružníc v každej farebnej triede bude 2-regulárny podgraf pôvodného grafu. Ak najmenší počet farieb potrebných na takéto zafarbenie je m potom ECD rozklad má *veľkosť* m . [2]



Obr. 1: ECD rozklad veľkosti 3 na červenú, modrú a zelenú kružnicu

Práca za semester: Do grafovej knižnice `ba-graph` som v jazyku `c++` naimplementoval backtracking (prehľadávanie s návratom) algoritmus, ktorý nájde veľkosť ECD grafu (súbor `ecd.hpp`). Pseudokód algoritmu je uvedený nižšie.

```
1: function NÁJDIECD(graf)
2:   inicializuj počet farebných tried na 0
3:   SKÚSZAČAĽNOVÚKRUŽNICU
4:   if žiadne ECD sa nenašlo then
5:     return neexistuje ECD
6:   else
7:     return najmenšie ECD
8:   end if
9: end function
10:
11: function SKÚSZAČAĽNOVÚKRUŽNICU
```

```

12:  if všetky hrany sú priradené farebnej triede then
13:      porovnaj veľkosť tohto ECD s doteraz najmenším nájdeným, ak je menšie zapamätaj si prira-
14:      denie hrán do farebných tried
15:      return
16:  else
17:       $h \leftarrow$  zober nejakú nepriradenú hranu
18:      for all  $f \in$  aktuálne farebné triedy do
19:          skús priradiť  $h$  do  $f$  a zapamätaj si, že  $h$  bude v kružnici na párnej pozícii
20:          SKÚSPOKRAČOVAŤVAKTUÁLNEJKRUŽNICI( $h$ )
21:      end for
22:      skús priradiť  $h$  do novej farebnej triedy a zapamätaj si, že  $h$  bude v kružnici na párnej pozícii
23:      if počet farebných tried je väčší alebo rovný ako aktuálne najmenšie nájdené ECD then
24:          odmietni aktuálne riešenie
25:          return
26:      end if
27:      SKÚSPOKRAČOVAŤVAKTUÁLNEJKRUŽNICI( $h$ )
28:  end if
29: end function
30: function SKÚSPOKRAČOVAŤVAKTUÁLNEJKRUŽNICI(aktuálnaHrana)
31:     ▷ Over či je aktuálnaHrana správne zafarbená. V korektnej ECD musia byť práve 2 susedné hrany
32:     v rovnakej farebnej triede a na pozíciách inej parity ako aktuálnaHrana ◁
33:      $p \leftarrow 0$            ▷ Počet susedných hrán v rovnakej farebnej triede ale na pozíciách rôznej parity
34:     for all  $h \in$  hrany susedné s aktuálnaHrana do
35:         if  $h$  je v rovnakej farebnej triede ako aktuálnaHrana then
36:             if  $h$  je na pozícii rovnakej parity ako aktuálnaHrana then
37:                 odmietni aktuálnu kružnicu
38:                 return
39:             else
40:                  $p \leftarrow p + 1$ 
41:             end if
42:         end if
43:     end for
44:     if  $p > 2$  then
45:         odmietni aktuálnu kružnicu
46:         return
47:     else if  $p = 2$  then
48:         SKÚSZAČAŤNOVÚKRUŽNICU           ▷ Bez sporu sa nám podarilo vytvoriť kružnicu párnej dĺžky
49:         return
50:     else
51:         for all  $h \in$  hrany susedné s aktuálnaHrana a aktuálne nepriradené do farebnej triedy do
52:             skús priradiť  $h$  do rovnakej farebnej triedy ako aktuálnaHrana ale na pozíciu inej parity
53:             SKÚSPOKRAČOVAŤVAKTUÁLNEJKRUŽNICI( $h$ )
54:         end for
55:     end if
56: end function

```

K algoritmu sú urobené aj testy (`testy_ecd.cpp`). Skúšal som viacero variant algoritmu (napríklad, že iterujeme cez všetky možné veľkosti ECD od najmenej po najväčšiu a pre každú veľkosť skúsime nájsť jedno zafarbenie. Ak ho nájdeme, skončíme.), ale tento bol najrýchlejší. Pri výbere hrany na riadku 16

máme viacej možností. Skúšal som program, kde je poradie výberu hrán náhodne a spustenia programu na rovnakých vstupoch trvali rôzne dlho. Teda to, v akom poradí sú tam vyberané hrany, ovplyvňuje rýchlosť. Skúšal som heuristiku, kde sa vždy vyberie hrana, ktorá má najviac susedov priradených do nejakej farebnej triedy. Rýchlejší bol však variant, kde sa vrcholy vyberajú na základe poradia, v ktorom sú prehľadávané do šírky BFS algoritmom od prvej hrany, ktorá sa zafarbí. Varianty backtrackingu sa dajú pozrieť ako rôzne `git` vetvy.

Práca na letný semester: ECD by sa malo dať nájsť pomocou SAT solvera.

Referencie

- [1] Reinhard Diestel. *Graph theory*. Springer (print edition); Reinhard Diestel (eBooks), 2024.
- [2] Analen A. Malnegro a Kenta Ozeki. „H-colorings for 4-regular graphs“. In: *Discrete Mathematics* 347.3 (2024), s. 113844. ISSN: 0012-365X. DOI: <https://doi.org/10.1016/j.disc.2023.113844>. URL: <https://www.sciencedirect.com/science/article/pii/S0012365X23005307>.