

# Zimný report

## Ciel' projektu:

(3, g, s)-multipól je 3-regulárny graf obvodu aspoň  $g$  s práve  $s$  polhranami. Polhrany možno ekvivalentne reprezertovať ako vrcholy stupňa 1. Obvod je dĺžka najkratšej kružnice. Cieľom projektu je vytvoriť počítačový program, ktorý pre zadané parametre nájde najmenší (3, g, s)-multipól. Pokiaľ to bude možné, program aj vygeneruje všetky takéto multipóly dostatočne malého rádu (počtu vrcholov). Očakáva sa, že algoritmus bude založený na efektívnom prehľadávaní s návratom a bude založený na matematickom poznaní týchto grafov.

## Práca vykonaná počas semestra:

- Oboznámenie sa s tému multipólov čítaním relevantných článkov
- Zoznámenie sa so základmi knižnice “Nauty and Traces” pre budúce použitie pri určovaní izomorfizmu grafov [1]
- Vytvorenie dátovej štruktúry vhodnú na reprezentovanie (3,g,s)-multipólov v C++
- Implementovanie pomocných algoritmov a funkcií pre generovanie binárnych podstromov
- Implementovanie testov pomocou CMake a CTest

## Štruktúra projektu:

### **Trieda AdjacencyListUndirectedGraph**

Je to trieda vhodná na reprezentovanie (3,g,s)-multipólov. Vrcholy sú očíslované, semiedge sa reprezentuje ako hrana do -1. Obsahuje metódy na pridanie vrcholov, hrán a semiedges, generovanie úplného binárneho stromu, taktiež metódy vhodné na testovanie (počet vrcholov, počet hrán, atď.).

### **Trieda RangeSet**

Pomocná trieda pre IntervalSkippingIterator, ktorá reprezentuje množinu odlišných intervalov čísel v rastúcom poradí.

### **Trieda IntervalSkippingIterator**

Iterátor, ktorý funguje rovnako ako napr. “for i in range(...)” v Pythone, len niektoré intervaly (reprezentované ako RangeSet) vynechá. Bude to pomocná funkcia pre ešte

neimplementovaný iterátor špecifických binárnych podstromov. Implementácia sa ešte bude meniť, lebo momentálne nemá operator overloading ako skutočné C++ iterátory.

## Plán na ďalší semester

Prvý krok: Implementovať iterátor, ktorý generuje všetky (alebo niektoré) podstromy s pevnou koreňou pre úplný binárny strom danej hĺbky pomocou dynamického programovania. To bude užitočné, lebo pre  $(3,g,s)$ -multipól existuje podstrom taký, že vnútorné vrcholy nemajú hrany medzi sebou – najväčšia možná hĺbka listu vyplýva z velkosti najmenšieho cyklu.

Druhý krok: Implementovať metódu ktorá doplní generované multipóly na  $(3,g,s)$ -multipóly ak je to možné.

## Citácie

1: McKay, B.D. and Piperno, A., Practical Graph Isomorphism, II, Journal of Symbolic Computation, 60 (2014), 94-112, <https://doi.org/10.1016/j.jsc.2013.09.003>