Report ročníkového projektu
Toky v sieťach
Adam Struharňanský

Naštudoval som si problematiku pomocou voľne dostupných zdrojov z internetu. Pomocou nej a rád od školiteľa som sa rozhodol na naprogramovanie nižšie uvedených algoritmov na ďalšie skúmanie. Všetky mnou naprogramované zdrojové kódy sú v C/C++ (vybrané kvôli rýchlosti).

Napísal som vlastnú implementáciu algoritmu Forda-Fulkersona, priamočiary algoritmus na daný problém.
Napísal som vlastné preformulovanie daného problému na problém lineárneho programovania. Tento bol pomocou súboru napísaného doc. RNDr. Robertom Lukoťkom, PhD. a pomocou solveru pre lineárne programovanie lp_solve (lp_solve) stiahnuteľným na http://lpsolve.sourceforge.net/5.5/ následne vyriešený. Dané preformulovanie funguje na princípe, že pre každú hranu musí mať množina všetkých ciest zo zdroja do ústia cez danú hranu menší alebo rovný spoločný prietok, ako má daná hrana kapacitu.
Našiel som implementáciu Dinicovho algoritmu (zdroj: https://cp-algorithms.com/graph/dinic.html , autori: https://github.com/e-maxx-eng/e-maxx-eng/commits/master/src/graph/dinic.md , bližšie informácie o implementácií priamo v súbore), ku ktorému som iba pridal metódu na zmenu jeho vstupu, tak ako je použité aj inde v programe.

Nakoniec boli všetky druhy prístupu skontrolované na náhodných grafoch, veľkosti 50 – 100 vrcholov, do 200 hrán.

---

lp_solve
lpsolve citation data

```
----------------------
Description     : Open source (Mixed-Integer) Linear Programming system
Language        : Multi-platform, pure ANSI C / POSIX source code, Lex/Yacc
based parsing
Official name   : lp_solve (alternatively lpsolve)
Release data    : Version 5.1.0.0 dated 1 May 2004
Co-developers   : Michel Berkelaar, Kjell Eikland, Peter Notebaert
Licence terms   : GNU LGPL (Lesser General Public Licence)
Citation policy : General references as per LGPL
                  Module specific references as specified therein
```