Report – Zimný semester
Vincent Hlaváč

Na implementáciu riešenia problému som sa rozhodol použiť programovací jazyk java.

Počas zimného semestra sa podarilo:

1. Zoznámiť so stable-tree dokompozíciami 4k+2-vrcholových kubických grafov.
2. Naprogramovať reprezentáciu neorientovaného grafu pomocou adjacency list (listu susednosti) – AdjacencyListGraph.java. Možnosť s ním pracovať pomocou univerzálneho grafového rozhrania – Graph.java.
3. Naprogramovať reader, ktorý zo súboru prečíta graf a vráti jeho objektovú reprezentáciu – AdjacencyListGraphReader.java. Možnosť s ním pracovať pomocou univerzálneho rozhrania pre grafové readere – GraphReader.java. V konštruktore mu možno špecifikovať kontrolu – GraphTypeControl.java, ktorú urobí na prečítanom grafe, ktorý vracia, len ak ju spĺňa. Implementoval som kontrolu pre 4k+2-vrcholový kubický graf – Cubic4KPlus2Control.java. Podarilo sa otestovať korektnosť readera.
4. Naprogramovať algoritmus generujúci všetky stable-tree dekompozície daného 4k+2-vrcholového kubického grafu – StableTreeDecomposition.java. Podarilo sa otestovať jeho korektnosť na základe porovnaní s očakávanými výstupmi (počty dekompozícii a počty výskytov jednotlivých vrcholov v ich stable setoch). Pre implementáciu algoritmu som naprogramoval repezentáciu komponentu grafu – Component.java.
5. Naprogramovať generátor konkrétnych grafov, ktorý vytvorý rozsiahlejšie grafy pre testovanie pri akom veľkom vstupe narazíme na exponenciálnu zložitosť daného backtrackového algoritmu. Konkrétne generujeme 4k+2-vrcholové kružnice s hranami medzi protiľahlými vrcholmi – Circle4KPlus2Generator.java a 2k+1-boké hranoly – Prism2KPlus1SidedGenerator. Možnosť s nimi pracovať pomocou rozhrania pre grafové generátory – GraphGenerator.java.

Všetky triedy sú súčasťou balíka graphs, ktorý tvorí akoby knižnicu jazyka java a možno ich používať vo vlastnom programe. Graph je možné vytvoriť len pomocou GraphReader, aby bolo zaručené, že vytvorený graf je korektný. StableTreeDecomposition zatiaľ umožňuje získať vytvorené stable-tree dekompozície len v podobe List<List<Integer>>, kde integre reprezentujú vrcholy zo stable setu dekompozície.

Ďalej som naprogramoval samostatné programy na testovanie a skúšobné použitie tried balíka graphs:

1. GraphGenerationTest – generuje súbory s grafmi na základe vstupných parametrov.
2. STDGenerationTest – načíta zo súboru určeného vo vstupných parametroch graf a vygeneruje jeho stable-tree dekompozície, ktoré uloží do súboru v provizórnom formáte výpisu popisujúcom vrcholy stable setu.

   *Popis vstupných parametrov je v komentáry v kóde.

Súborová reprezentácia grafu:

Na prvom riadku je číslo n reprezentujúce počet vrcholov grafu. Ďalších n riadkov prislúcha postupne všetkým vrcholom s indexami od 0 po n-1. Na riadku prislúchajúcemu vrcholu s indexom i sú indexy tých vrcholov do ktorých ide z i hrana a zároveň sú vačšie než i. Takto popisujeme každú hranu len raz. Súbor nesmie obsahovať žiadne prebytočné neblakové znaky.