Fakulta riadenia a informatiky Informatika

Semestrálna práca

*Diskrétna optimalizácia*

Zadanie H6

Letný semester 2019 Jakub Senko 5ZY028

Obsah

[Zadanie 3](#_Toc7945761)

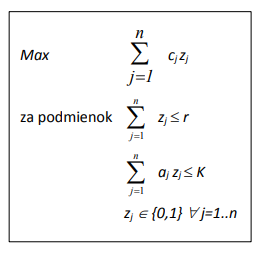
[Popis riešeného algoritmu 4](#_Toc7945762)

[Popis jednotlivých tried 6](#_Toc7945763)

[Záverečné zhodnotenie 6](#_Toc7945764)

# Zadanie

* Duálnou heuristikou s koeficientmi výhodnosti riešite úlohu danú modelom (úloha o batohu s kapacitou K a obmedzeným počtom predmetov v batohu r).
* Riešte úlohu pre:
  + **n = 500** (počet všetkých predmetov)
  + **r = 300** (maximálny počet predmetov v batohu)
  + **K = 15000** (nosnosť batohu - kapacita)
* a pre lokálne kritérium „Odstráň prvok z dosiaľ nespracovaných prvkov, ktorý má najmenší pomer koeficientov (najmenší výhodnostný koeficient)“.
* Východiskové riešenie položte rovné batohu, v ktorom sú vložené všetky prvky (neprípustné riešenie).
* Súčasťou zadania sú súbory H6\_a.txt a H6\_c.txt, ktoré obsahujú n údajov koeficientov a pre j = 1..n potrebných pre riešenie zadanej úlohy
* Zápis účelovej funkcie a štrukturálnych podmienok:



# Popis riešeného algoritmu

Algoritmus je celý riešený v metóde main v triede Main.

Pred začiatkom sú zadefinované atribúty:

* Kapacita = 15000 (podľa zadania)
* pocetPredmetov = 300 (podľa zadania)
* polia
  + ceny (účelová funkcia – načítané ceny zo súboru H6\_c.txt)
  + hmot (štrukturálne podmienky – načítané hmotnosti zo súboru H6\_a.txt)
  + ries (obsahuje 0/1, false/true – či bol predmet pridaný do batohu)

Začína načítaním dát zo súboru cez pomocnú triedu FileManager na spracovanie súborov.

Následne pokračuje výpočtom koeficientov výhodnosti a ich usporiadaním od najmenšieho po najväčší do údajovej štruktúry TreeMap<Integer, Double> kde Integer slúži na indexovanie a Double na uloženie hodnoty koeficientu výhodnosti. Algoritmus pokračuje usporiadaním koeficientov výhodnosti cez údajovú štruktúru LinkedList a Comparator, ktorá sa usporiada pomocou metódy sort zo základnej knižnice Java. Po usporiadaní sa inicializuje pole ries, ktoré bude obsahovať riešenie heuristiky vo forme boolean kde true = predmet je v batohu, a false = predmet nie je v batohu. Batoh sa nastaví plný kvôli zadaniu úlohy (nastaví sa do neprípustného riešenia), čiže počet prvkov v batohu bude rovný 500 a hmotnosť batohu bude vypočítaná podľa zadania v súbore (29569). Po naplnení batohu sa vykoná cyklus while, ktorý bude pokračovať dovtedy, pokým nebude batoh v prípustnom riešení. Po vykonaní cyklu sa vypíše výsledok heuristiky na terminál a do súboru „vystup“ v zložke „vystupne subory“.

# Popis jednotlivých tried

Pre spustenie programu je potreba najnovšej Javy (JDK11).

Triedy sú podrobnejšie zdokumentované v samotnom kóde (keďže dokumentovať triedy vo Worde je dosť nepraktické), alebo aj v [JavaDoc](JavaDoc/index.html) vygenerovanom súbore, ktorý si môžete otvoriť v prehliadači.

## FileManager

Pomocná trieda z iných mojich projektov, ktorá umožňuje načítať súbory do matice, z ktorými sa dá následne pracovať.

## Main

Trieda obsahujúca všetko ohľadom daného algoritmu.

# Záverečné zhodnotenie

Výsledné riešenie zadania:

Predmety, ktoré boli pridané do batohu sú vypísané v súbore „vystup.txt“ v zložke projektu: vystupne subory

Hodnota účelovej funkcie: 32250

Výsledná hmotnosť: 14995

Počet predmetov: 268