Praca domowa 02 – hashing

Michał Szczygieł

Wstęp

Zadanie składa się z 2 plików:

- Hash.java
- Hashing.java

Klasa Hashing

Klasa ta odpowiedzialna jest za wczytanie danych z pliku, oraz przekazanie poprawnie sofrmułowanych danych do klasy Hash.

Metody:

`private static LinkedList<String> readHashFile(File file)` - Czytanie danych z pliku wejściowego oraz zwrócenie wyniku w postaci listy stringów.

`private static void chaining(Hash hash, LinkedList<String> list)` – Dodawanie stringów z pliku do hashowania.

Klasa Hash

Klasa Hash oblicza ilość kolzji poprzez wstawianie elementu do listy hashy. Oprócz tego klasa, realizuje podstawowe metody takie jak wstawianie, szukanie i usuwanie.

Pola:

`private int collisionAmount = 0;` - Zmienne przechowywująca ilość kolizji

`private ArrayList<LinkedList<String>> hashTable;` - Kolekcja przechowywująca listę hashy.

Metody:

```
public Hash(int hashSize)

private int binarySearch(LinkedList<String> list, String word)

public boolean find(String word)

public int getCollisionAmount()

private LinkedList<String> getHash(String word)

public void insert(String word)

public void remove(String word)
```

Realizacja algorytmu wstawiania hashy

Algorytm wstawia kolejne słowa z pliku na poczatek listy, oraz dokonuje ich hashowania. Jeśli algorytm naptokta nie pustą listę przeszkuje listę w sposób binarny. Do Hashowania wykorzystałem methodę hashCode() - 32bitowa. Wykorzystanie przeszukiwania binarnego ma na celu zmniejszenia ilości kolizji, a jego złożoność obliczeniowa wynosi O(n log n).

Dyskusja

Problem może być potrakotwany bardziej w sposób złożóny poprzez zastosowanie lepszej funkcji skrótu np. SHA-1 (160bitowej). I dla dużego zbioru (hash size) liczb ilość kolizji automatycznie powinna spaść.