Lista 4

Artur Pazurkiewicz

Założenia odnośnie programu:

- ‘A’ < ‘a’
- klucze nie muszą być unikalne

Sposób uruchomienia programu:

- należy udać się do ścieżki **lista4/target**
- program uruchamia się po wpisaniu komendy:
  java -jar lista4-1.0.jar
- do uruchomienia wymagane jest podanie przynajmniej -i nazwaPliku w którym znajdują się po kolei komendy
- **java -jar lista4-1.0.jar -h** wyświetla sposoby na uruchomienie wraz z możliwymi argumentami do wpisania

Możliwości programu (wyjaśnienie mode):

- **--mode normal** (domyślnie) program zgodnie z poleceniem wczytuje komendy z pliku i wyświetla informacje na wyjście błędu
- **--mode test1** program wczytuje komendy z pliku, wykonuje z niego tylko load oraz insert. Następnie wykonuje dla każdego elementu w pliku przeszukanie. Na podstawie czasu który zajęły obie operacje, program sortuje wyniki po atrybucie będącym sumą 20*czas przeszukiwania oraz czas wstawiania.
  - Za pomocą --min oraz --max można ustalić zakres przeszukiwanego nt
  - -k definiuje ilość prób
- **--mode test2** program wczytuje komendy z pliku, wykonuje z niego tylko load oraz insert. Następnie wywołuje wielokrotnie funkcję find (szuka słów włożonych w strukturę) i na tej podstawie oblicza minimalną, maksymalną oraz średnią wartość.
- Daje nam to podstawy do weryfikacji teorii odnośnie górnego oraz dolnego ograniczenia

Zadanie 1

- Zależnie od wielkości danych, różne nt daje różne rezultaty.
- --mode test1

Zadanie 2

- Wszystkie tezy które postawię w tym zadaniu można łatwo zweryfikować przy użyciu --mode test2
- n = ilość elementów
- Licząc ilość porównań kluczowy jest aspekt wysokości drzewa. Głębokość danego elementu w strukturze definiuje ile porównań trzeba wykonać
  - Tezy odnośnie górnego ograniczenia (java -jar lista4-1.0.jar -i test2.txt --mode test2 --type wybranyTyp):
    - BST
      - Górne ograniczenie wynosi $O(n)$, gdy elementy będą uporządkowane

        - Spowodowane jest to sposobem dodawania nowych elementów
    - RBT
        - Górne ograniczenie wynosi $O(\log_2 n)$
    - HMAP
        - Ponieważ HashMapa zawiera w swojej implementacji RBT, gdy wszystkie elementy pójdą do jednego koszyka, górne ograniczenie będzie wynosić $O(\log_2 n)$
  - Tezy odnośnie dolnego ograniczenia (java -jar .\lista4-1.0.jar -i .\test.txt --mode test2 --type wybranyTyp)
    - Dolne ograniczenia dla BST, RBT i HMAP wynosi $\Omega(1)$
- Tezy odnośnie średniej:
  - BST
    - Średni dostęp do elementu znacząco zależy od kolejności wkładania elementów do struktury
  - RBT
    - Średni dostęp do elementu wynosi $O\left(\log_2 \frac{n}{2}\right) = O(\log_2 n)$
  - HMAP
    - Średni dostęp do elementu wynosi $O\left(\log_2 \frac{n}{2m}\right) = O(\log_2 n)$

Przykładowe wywołania:

- java -jar lista4-1.0.jar -i command.txt -m 3 --type hmap -nt 10
- java -jar lista4-1.0.jar -t rbt -i test.txt
- java -jar lista4-1.0.jar -t bst -i test.txt
- java -jar lista4-1.0.jar -i command.txt --mode test1
- java -jar lista4-1.0.jar -i test.txt --mode test2 -nt 10 -t rbt
- java -jar lista4-1.0.jar -i test.txt --mode test1 --min 10 --max 20