

README

Lista 4

Jakub Gogola

12 maja 2018

W niniejszym katalogu zamieściłem testy wykonane przeze mnie w ramach listy 4. Poniżej krótko przedstawię ich wyniki.

Pierwszym wykonanym testem było wyznaczenie rozmiaru struktury, dla którego opłaca się w tablicy hashującej przełączyć z listy na drzewo czerwono-czarne. Wykonałem test, w którym dla różnej ilości danych (od 20 do 900 słów, shift co 20) wczytywałem jednocześnie do listy i RB-Drzewa te same dane i następnie wyszukiwałem w nich dane położone na losowych indeksach. Wyniki tego testu pokazały, że takie przełączenie powinno nastąpić w momencie, gdy struktura osiągnie rozmiar 300 elementów. Dane wynikowe z tego testu znajdują się w pliku "nt.res".

Kolejną sprawdzaną przeze operacją był średni czas wykonywania operacji na strukturach (wyniki w pliku "ot.res"). Najlepiej wypadło drzewo czerwono-czarne, później była tablica hashująca, a na końcu drzewo poszukiwań binarnych.

Ostatnia sprawdzana rzecz to liczba porównań wykonywanych przez struktury przy operacji find (zadanie 2. z listy 4.). W teście wczytywałem do struktur fragment Biblii (przykład tekstu z powtarzającymi się słowami) oraz słownik (tekst bez powtórzeń). Oto wyniki (znajdują się one również w pliku "on.res"):

```
With repeats -> bible.txt
```

```
BST:
```

```
~~~Min: 9
```

```
~~~Avg: 45
```

```
~~~Max: 159
```

```
RBT:
```

```
~~~Min: 1
```

```
~~~Avg: 8
```

```
~~~Max: 16
```

```
HT:
```

```
~~~Min: 0
```

```
~~~Avg: 8
```

```
~~~Max: 44
```

```
Without repeats -> aspell_wordlist.txt
```

```
BST:
```

```
~~~Min: 715
```

```
~~~Avg: 2018
```

```
~~~Max: 3767
```

```
RBT:
```

```
~~~Min: 9
```

```
~~~Avg: 12
```

```
~~~Max: 16
```

```
HT:
```

```
~~~Min: 14
```

```
~~~Avg: 78
```

```
~~~Max: 165
```

W każdym teście wczytywałem do struktur jednocześnie po 20 tysięcy słów. Następnie robiłem po 20 powtórzeń dla każdej, gdzie usiłowałem wyszukać losowy element w strukturze. W przypadku obu rodzajów tekstów najgorzej wypadło BST, a najlepiej RBT. Da się zauważyć, że w

przypadku, gdy w zbiorze wyrazów mamy do czynienia z powtórzeniami, liczba wykonywanych porównań w każdym przypadku jest mniejsza. Wynika to z tego, że jest duże prawdopodobieństwo, iż w strukturze występuje kilka takich samych słów. Najgorsze wyniki osiąga drzewo BST, ponieważ nie jest ono zbalansowane (złożoność operacji wyszukiwania wynosi $\mathcal{O}(h)$), gdzie h jest wysokością drzewa. Najszybciej wyszukiwanie odbywa się w przypadku drzewa czerwono-czarnego, gdzie złożoność tej operacji to $\mathcal{O}(\log n)$ (n to liczba węzłów w drzewie). Wpływ na liczbę porównań w tablicy hashującej ma fakt czy na danej pozycji tablicy znajduje się aktualnie lista czy drzewo RBT.

Przyglądając się wynikom dla zbioru słów bez powtórzeń, widzimy, że "osiągi" drzewa czerwono-czarnego się praktycznie nie zmieniły (wynika to z jego zbalansowania), dla tablicy hashującej nieco się pogorszyły, a dla drzewa poszukiwań binarnych liczba wykonywanych porównań drastycznie wzrosła. Wynika to z tego, że BST jest drzewem niezbalansowanym (w przeciwieństwie do RBT czy tablicy hashującej, w której jest spore prawdopodobieństwo, że na danej jej pozycji znajduje się aktualnie RBT).