

“Drzewo binarne, tablica haszujaca”

Martyna Bandura

09 kwietnia 2014

1. Wstęp:

Celem ćwiczenia jest przetestowanie czasu implementacji takich struktur danych jak:

- drzewo binarne poszukiwań;
- tablica haszująca.

2. Opis wykorzystanych algorytmów:

• *Drzewo binarne poszukiwań:*

- Drzewo poszukiwań binarnych BST (ang. Binary Search Tree) jest dynamiczną strukturą danych zbudowaną z węzłów (ang. node). Każdy węzeł może posiadać dwóch potomków (left - lewy i right - prawy) oraz jednego przodka (p - parent). Z każdym węzłem dodatkowo związany jest klucz (key).
- Dla każdego węzła w drzewie BST zachodzą następujące własności:
 - * Wartości kluczy węzłów leżących w lewym poddrzewie węzła są mniejsze lub równe wartości klucza danego węzła.
 - * Wartości kluczy węzłów leżących w prawym poddrzewie węzła są większe lub równe wartości klucza danego węzła.

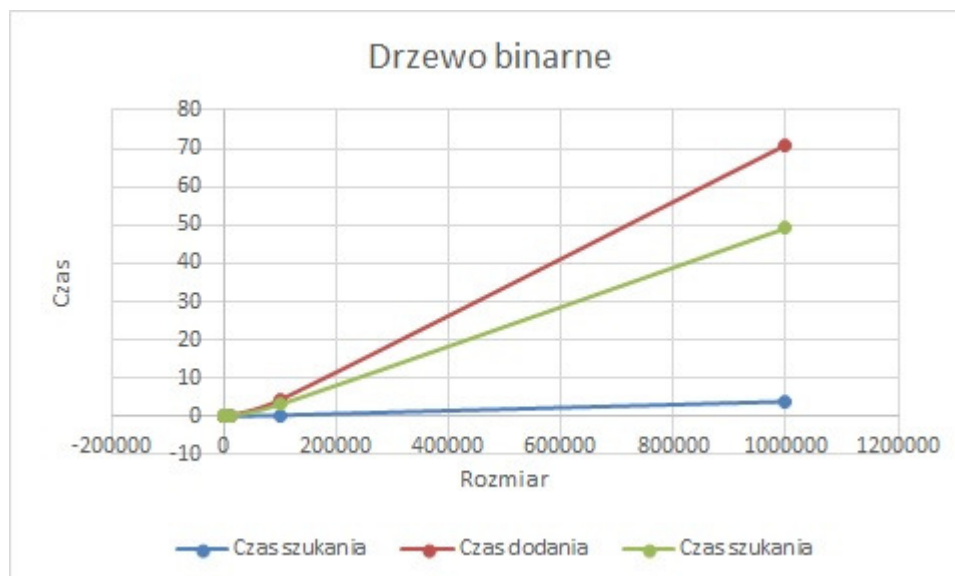
• *Tablica haszująca:*

- Tablice haszujące stosuje się do zmniejszenia czasu przeszukiwania i dostępu do danych zawartych w strukturze liniowej jak np. lista czy tablica. Zbiór danych dzieli się na rozłączne podzbiory, których adresy utożsamiane są z poszczególnymi komórkami tablicy haszującej h. To, z jakim podzbiorem danych związana jest komórka tablicy h[k] określa się za pomocą tzw. funkcji haszującej. Funkcję tą określamy na całym dopuszczalnym zbiorze danych, jej wartością jest indeks w tablicy haszującej, który wyznacza niejako pewien podzbiór, do którego powinien należeć poszukiwany element zbioru danych.

3. Testy wydajności:

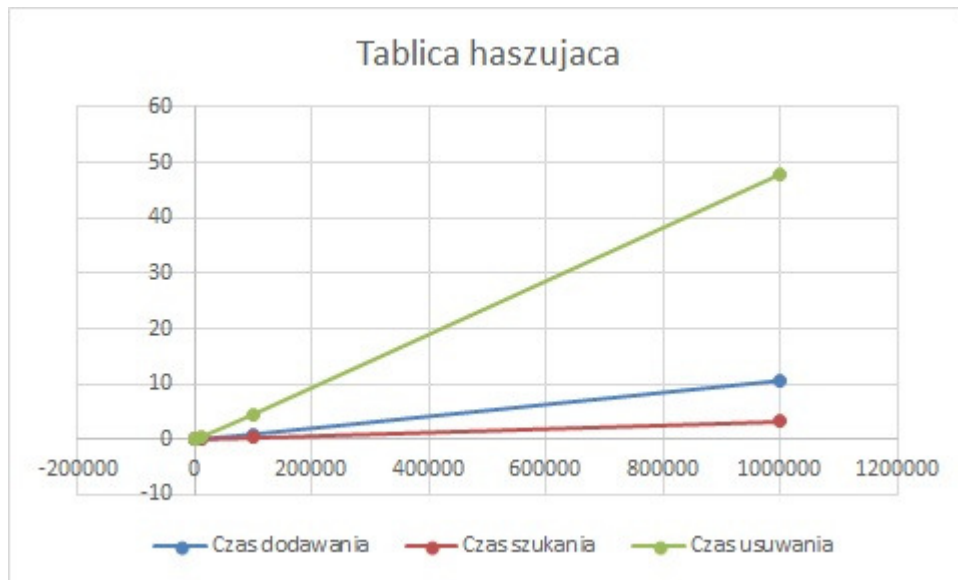
- Drzewo binarne:

| Rozmiar | Czas dodania | Czas szukania | Czas usuwania |
|---------|--------------|---------------|---------------|
| 10 | 0 | 0 | 0 |
| 100 | 0,003 | 0 | 0,01 |
| 1000 | 0,028 | 0,003 | 0,013 |
| 10000 | 0,356 | 0,031 | 0,188 |
| 100000 | 4,372 | 0,363 | 3,152 |
| 1000000 | 70,941 | 3,899 | 49,165 |



- Tablica haszująca:

| N | Czas dodawania | Czas szukania | Czas usuwania |
|---------|----------------|---------------|---------------|
| 10 | 0 | 0 | 0 |
| 100 | 0 | 0,001 | 0,001 |
| 1000 | 0,009 | 0,002 | 0,02 |
| 10000 | 0,093 | 0,025 | 0,42 |
| 100000 | 0,938 | 0,262 | 4,537 |
| 1000000 | 10,616 | 3,159 | 47,892 |



Wnioski:

- Algorytmy cechują się dużą szybkością. Dla danych, które są rzędów milionowych czas wyszukiwań jest odpowiedni.
- Drzewo binarne posiada złożoność obliczeniową $O(\log_2 n)$. Istnieje jednak ryzyko złożoności pesymistycznej $O(n)$.
- Tablica haszująca posiada złożoność obliczeniową równą $O(1)$. Można to zauważyć podczas testów dla rzędów powyżej miliona. Przypadek pesymistyczny wynosi $O(n)$.
- Tablica haszująca jest o wiele szybszym przypadkiem niż drzewo binarne. Najbardziej zauważalne jest to przy większych rozmiarach problemu.