**POLITECHNIKA ŚWIĘTOKRZYSKA W KIELCACH**

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI I INFORMATYKI**

Kierunek studiów**: Informatyka, rok I**

Przedmiot**: Algorytmy i struktury danych – projekt, semestr 2**

Temat projektu**: Drzewo BST - wstawianie, usuwanie, wyszukiwanie**

Autor**: Arkadiusz Więcław**

**1. Treść zadania**

Projekt polega na przeanalizowaniu algorytmów wstawiania, usuwania i wyszukiwania w drzewie BST. Zmieniając liczbę danych (przy losowym uporządkowaniu) i uporządkowanie danych (przy stałej ich liczbie) w programie należało zliczyć liczbę kluczowych operacji, przeanalizować wyniki i wyciągnąć wnioski.

**2. Opis działania programu**

Aplikacja została napisana w systemie operacyjnym Microsoft Windows przy pomocy zintegrowanego środowiska programistycznego (IDE) CodeBlocks. Użyto języka programowania C++. Szkielet aplikacji został oparty na interfejsie graficznym z wykorzystaniem biblioteki graficznej OpenGL oraz GLUT 3.7.

Program pokazuje działanie algotymów wstawiania, usuwania i wyszukiwania w drzewie BST. Wyniki prezentowane są w formie graficznej – widać zmiany w strukturze drzewa BST po ukończeniu wybranego algorytmu.

Użytkownik po naciśnięciu prawego przycisku myszy ma następujące możliwości:

* dodanie nowego węzła,
* wyszukanie węzła o podanej wartości klucza,
* wyczyszczenie wyników wyszukiwania,
* usuwanie węzła,
* generowanie nowego drzewa losowo,
* generowanie nowego drzewa rosnąco,
* generowanie nowego drzewa malejąco,
* zakończenie programu.

**3. Wyniki przeprowadzonej analizy algorytmów**

**3.1. Losowe uporządkowanie**

Poniżej przedstawiono wyniki analizy działania algorytmów dla losowego uporządkowania.

Tabela 1. Liczba operacji przy losowym uporządkowaniu elementów

|  |  |  |  |
| --- | --- | --- | --- |
| Liczba węzłów | Liczba porównań | Liczba przebiegów pętli | Liczba przypisań |
| 10 | 61 | 24 | 133 |
| 20 | 215 | 88 | 357 |
| 30 | 295 | 118 | 507 |
| 40 | 429 | 181 | 711 |
| 50 | 604 | 260 | 956 |
| 60 | 727 | 310 | 1149 |
| 70 | 814 | 360 | 1306 |
| 80 | 1109 | 481 | 1671 |
| 90 | 1213 | 529 | 1845 |
| 100 | 1330 | 591 | 2032 |

Rys. 1. Graficzne przedstawienie tabeli 1.

Z wykresu wynika, że przy losowym uporządkowaniu liczba operacji przy wstawianiu elementów do drzewa rośnie liniowo.

**3.1. Rosnące i malejące uporządkowanie elementów**

Poniżej przedstawiono wyniki analizy działania algorytmów dla rosnącego i malejącego uporządkowania.

Tabela 2. Liczba operacji przy rosnącym uporządkowaniu elementów

|  |  |  |  |
| --- | --- | --- | --- |
| Liczba węzłów | Liczba porównań | Liczba przebiegów pętli | Liczba przypisań |
| 10 | 109 | 45 | 181 |
| 20 | 419 | 190 | 561 |
| 30 | 929 | 435 | 1141 |
| 40 | 1639 | 780 | 1921 |
| 50 | 2549 | 1225 | 2901 |
| 60 | 3659 | 1770 | 4081 |
| 70 | 4969 | 2415 | 5461 |
| 80 | 6479 | 3160 | 7041 |
| 90 | 8189 | 4005 | 8821 |
| 100 | 10099 | 4950 | 10801 |

Rys. 2. Graficzne przedstawienie tabeli 2.

Z wykresu wynika, że przy rosnącym uporządkowaniu liczba operacji przy wstawianiu elementów do drzewa rośnie wykładniczo.

Tabela 3. Liczba operacji przy malejącym uporządkowaniu elementów

|  |  |  |  |
| --- | --- | --- | --- |
| Liczba węzłów | Liczba porównań | Liczba przebiegów pętli | Liczba przypisań |
| 10 | 109 | 45 | 181 |
| 20 | 419 | 190 | 561 |
| 30 | 929 | 435 | 1141 |
| 40 | 1639 | 780 | 1921 |
| 50 | 2549 | 1225 | 2901 |
| 60 | 3659 | 1770 | 4081 |
| 70 | 4969 | 2415 | 5461 |
| 80 | 6479 | 3160 | 7041 |
| 90 | 8189 | 4005 | 8821 |
| 100 | 10099 | 4950 | 10801 |

Rys. 3. Graficzne przedstawienie tabeli 3.

Z wykresu wynika, że przy malejącym uporządkowaniu liczba operacji przy wstawianiu elementów do drzewa rośnie wykładniczo, czyli tak samo jak w przypadku rosnącego uporządkowania liczb.

**3.3. Porównanie liczby operacji dla różnych uporządkowań**

Poniżej przedstawiono porównanie liczby operacji przy wstawianiu elementów do drzewa dla różnych uporządkowań. Analizę przeprowadzono dla liczby węzłów z zakresu od 10 do 100.

Rys. 4. Liczba porównań w zależności od uporządkowania

Rys. 5. Liczba przypisań w zależności od uporządkowania

Rys. 6. Liczba przebiegów pętli w zależności od uporządkowania

Z wykresów wynika, że wraz ze wzrostem liczby węzłów w drzewie BST zwiększa się różnica między liczbą operacji przy uporządkowaniu losowym a liczbą operacji przy uporządkowaniu rosnącym/malejącym.

**4. Wnioski**

Przy losowym uporządkowaniu węzłów w drzewie BST liczba wykonywanych operacji jest znacznie mniejsza niż w przypadku uporządkowania rosnącego lub malejącego. Warto więc dbać o to aby drzewo było dobrze wyważone wówczas dostęp do danych jest znacznie szybszy. Przy rosnącym uporządkowaniu drzewo przyjmuje postać listy, a dostęp do ostatnich węzłów jest bardzo czasochłonny. Z tego wynika, że drzewo BST jest bardziej wydajne od innych niedrzewiastych struktur danych.