**Sprawozdanie**

**Metody dostępu do danych**

Zad. 2

Wprowadzić do tablicy dane daneA i wyszukać daneB oraz daneC. Porównać wyniki prób wyszukania metodą przeszukiwania sekwencyjnego dla danych przypadkowych, dla danych posortowanych.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Zad 2 | | | | | | | Zad 3 | | | | | |
| Wyszukiwanie | B | | | C | | | B | | | C | | |
|  | Min | Max | Śr | Min | Max | Śr | Min | Max | Śr | Min | Max | Śr |
| Nieposortowane | 1 | 97 | 48,2 | 1 | 100 | 49,22 | 3 | 7 | 5,86 | 3 | 7 | 5,8 |
|  | Min | Max | Śr | Min | Max | Śr | Wyszukiwanie metodą podziałów dychotomicznych | | | | | |
| Posortowane | 3 | 100 | 50,02 | 3 | 100 | 49,54 |
| Wnioski: Średnia liczba wyszukań danych nieposortowanych jest nieznacznie mniejsza niż tych posortowanych. | | | | | | | Wnioski: W przypadku wyszukiwania metodą podziałów dychotomicznych można zauważyć znaczny spadek liczby zapytań maksymalnych jak i średnich w porównaniu do metody sekwencyjnej. Jest ona szybsza od metody z poprzedniego zadania. | | | | | |

Zad. 3

Te same testy przeprowadzić dla wyszukiwania metodą podziałów dychotomicznych.

Zad. 4

Przeprowadzić eksperymenty z drzewami binarnymi. Porównać uzyskane wyniki dla „zwykłych” drzew (bez wyważenia) i drzew dokładnie wyważonych.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Zad. 4 | **B** | | | C | | | | Drzewo binarne | Min | Max | Śr | Min | Max | Śr | | 1 | 12 | 7 | 1 | 12 | 6,56 | | Drzewo dokładnie wyważone | Min | Max | Śr | Min | Max | Śr | | 3 | 7 | 5,86 | 3 | 7 | 5,8 | | Wnioski: Średnia liczba wyszukań dla drzew dokładnie wyważonych jest mniejsza w stosunku do drzew bez wyważania. Maksymalna liczba zapytań jest zdecydowanie większa dla drzew bez wyważania w porównaniu z drzewami dokładnie wyważonymi. Wywnioskować można, że wyszukiwanie po drzewie dokładnie wyważonym jest szybsze. |

Zad. 5

Określić kolejność wprowadzania do drzewa 31 liczb (od 1 do 31) tak by uzyskać dokładnie wyważone drzewo binarne.

16, 8, 24, 4, 12, 20, 28, 2, 6, 10, 14, 18, 22, 26, 30, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31

Zad. 6

Obliczyć minimalną i maksymalną liczbę elementów jakie można umieścić w B-drzewie klasy t(h,m).

Zad. 7

Przeprowadzić eksperymenty dotyczące wyszukiwania w B-drzewach, dla różnych wielkości strony. Określić optymalny ze względu na czas wyszukiwania rozmiar strony.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Zad. 7 | B | | | | | | | | | | | C | | | | | | | | | | | |
| Rozmiar strony M | m=2 | | | m=3 | | | | m=5 | | | | m=2 | | | | m=3 | | | | m=5 | | | |
|  | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | |
| Dostępy do pamięci | 3 | 8 | 6,18 | | 2 | 8 | 5,8 | | 2 | 8 | 5,82 | | 2 | 8 | 5,64 | | 2 | 7 | 5,34 | | 2 | 8 | 5,34 | |
|  | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | |
| Dostępy do dysku | 2 | 4 | 3,74 | | 1 | 3 | 2,7 | | 2 | 3 | 2,8 | | 2 | 4 | 3,48 | | 1 | 3 | 2,58 | | 2 | 3 | 2,74 | |

Wnioski: Na podstawie powyższej tabeli można wywnioskować, że pod względem dostępów do pamięci oraz ilości dostępów do dysku najlepszym wyborem jest rozmiar strony równy 3, chociaż niewiele gorszym rozwiązaniem jest rozmiar strony równy 5. Najgorzej wypada m=2.

Zad. 8

Te same testy przeprowadzić dla wyszukiwania w B\*-drzewach. Określić optymalny ze względu na czas wyszukiwania rozmiar strony.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Zad. 8 | B | | | | | | | | | | | C | | | | | | | | | | | |
| Rozmiar strony M | m=2 | | | m=3 | | | | m=5 | | | | m=2 | | | | m=3 | | | | m=5 | | | |
|  | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | |
| Dostępy do pamięci | 5 | 9 | 6,9 | | 5 | 8 | 6,64 | | 4 | 8 | 6,46 | | 5 | 9 | 7,38 | | 5 | 8 | 6,54 | | 4 | 8 | 6,3 | |
|  | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | | Min | Max | Śr | |
| Dostępy do dysku | 4 | 4 | 4 | | 3 | 3 | 3 | | 3 | 3 | 3 | | 4 | 4 | 4 | | 3 | 3 | 3 | | 3 | 3 | 3 | |

Wnioski: Z powyższych wyników można wywnioskować, że pod względem ilości dostępów do pamięci dla podanego zakresu liczb najlepszym rozwiązaniem jest rozmiar strony równy 5, gdzie średni czas dla zbioru B wynosi 6,46 a dla zbioru C 6,3. Jeśli chodzi o ilość dostępów do dysku to zarówno m=3 i m=5 są równo efektywne, gdzie dla każdego zbioru danych średnia ilość to 3.

Zad. 9

Porównać wyniki uzyskane dla B-drzew i B\*-drzew.

B-drzewo posiada korzystniejsze wyniki dostępu do dysku oraz pamięci. Wyszukiwanie metodą B-drzewa jest wydajniejsze.

Zad.10

Przeprowadzić eksperymenty dotyczące wstawiania danych do tablicy mieszającej. Sprawdzić wpływ rozmiaru tablicy, wybranej funkcji mieszającej i funkcji rozwiązywania kolizji na liczbę kolizji występujących podczas wstawiania.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Wielkość tablicy 150 | Dzielenie przez rozmiar tablicy | | | Podział, składanie i dzielenie | | | Wycięcie 3 cyfr klucza i normalizacja | | |
| Sondowanie liniowe z krokiem 1 | Min | Max | Śr | Min | Max | Śr | Min | Max | Śr |
| 1 | 13 | 2,04 | 1 | 17 | 4,17 | 1 | 16 | 4,37 |
| Sondowanie liniowe z krokiem 7 | Min | Max | Śr | Min | Max | Śr | Min | Max | Śr |
| 1 | 7 | 1,89 | 1 | 8 | 2,81 | 1 | 8 | 2,61 |
| Podwójne mieszanie zależne | Min | Max | Śr | Min | Max | Śr | Min | Max | Śr |
| \* | \* | \* | 1 | 13 | 3,88 | \* | \* | \* |

Wnioski: Jeśli wielkość tablicy jest niewiele większa od ilości wprowadzanych danych to nie możliwe jest skorzystanie z funkcji podwójnego mieszania zależnego dla funkcji dzielenia przez rozmiar tablicy oraz wycięciu 3 cyfr klucza z normalizacją.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Wielkość tablicy 200 | Dzielenie przez rozmiar tablicy | | | Podział, składanie i dzielenie | | | Wycięcie 3 cyfr klucza i normalizacja | | |
| Sondowanie liniowe z krokiem 1 | Min | Max | Śr | Min | Max | Śr | Min | Max | Śr |
| 1 | 5 | 1,36 | 1 | 8 | 2,15 | 1 | 7 | 2,3 |
| Sondowanie liniowe z krokiem 7 | Min | Max | Śr | Min | Max | Śr | Min | Max | Śr |
| 1 | 6 | 1,46 | 1 | 8 | 1,91 | 1 | 8 | 2,14 |
| Podwójne mieszanie zależne | Min | Max | Śr | Min | Max | Śr | Min | Max | Śr |
| 1 | 2 | 1,25 | 1 | 2 | 1,53 | 1 | 2 | 1,6 |

Wnioski: Zwiększenie wielkości tablicy do 200 spowodowało obniżeniem ilości kolizji w porównaniu z poprzednią wielkością. Dodatkowo nie mamy dla tej wielkości tablicy nieskończonej liczby kolizji w przypadku wykorzystania funkcji podwójnego mieszania zależnego.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Wielkość tablicy 300 | Dzielenie przez rozmiar tablicy | | | Podział, składanie i dzielenie | | | Wycięcie 3 cyfr klucza i normalizacja | | |
| Sondowanie liniowe z krokiem 1 | Min | Max | Śr | Min | Max | Śr | Min | Max | Śr |
| 1 | 7 | 1,8 | 1 | 6 | 1,72 | 1 | 4 | 1,26 |
| Sondowanie liniowe z krokiem 7 | Min | Max | Śr | Min | Max | Śr | Min | Max | Śr |
| 1 | 5 | 1,48 | 1 | 4 | 1,45 | 1 | 4 | 1,24 |
| Podwójne mieszanie zależne | Min | Max | Śr | Min | Max | Śr | Min | Max | Śr |
| 1 | 2 | 1,37 | 1 | 2 | 1,35 | 1 | 2 | 1,17 |

Wnioski: Przy zwiększeniu wielkości tablicy do 300 średnie ilości kolizji ulegają zmniejszeniu. Wadą takiego postępowania jest marnotrawienie dostępnych zasobów w przypadku ustawiania coraz to większych wielkości tablic.

\*- nieskończona liczba kolizji

Podsumowanie: Najmniejszą ilość kolizji zapewnia nam podwójne mieszanie zależne, jednak nie jest ono możliwe, jeśli wielkość tablicy nie zapewnia nadmiarowych miejsc. Ciągłe zwiększanie wielkości tablicy powodowałoby zmniejszenie średnich ilości kolizji, jednak prowadzi to do marnotrawstwa zasobów.