

Wykonawca: Sebastian Franczak							
<b>Laboratorium Podstaw Informatyki</b>							
Temat:	Metody dostępu do danych					Nr.lab.: 5	
Rok akademicki	termin	Rodz. studiów	grupa	Data lab.	Data oddania sprawozdania	prowadzący	ocena
2011/2012 semestr letni		dziennie	2	31.05	14.06	Robert Tutajewicz	

1)

Przygotowano zbiór z danymi w skład których wchodzi 200 wygenerowanych losowo liczb. Wśród nich szukać będziemy 100 liczb.

63	33	524	244	374
71	219	785	623	674
927	691	93	251	991
298	493	358	523	459
67	621	237	25	481
899	621	280	919	503
417	511	298	0	57
145	111	544	170	696
465	570	481	552	689
609	59	637	971	165
47	152	646	49	959
40	604	427	853	535
354	644	274	722	404
604	318	775	353	788
112	179	568	458	557
609	752	119	616	156
366	275	695	197	
288	990	405	373	
648	398	985	524	
864	586	775	440	
515	328	18	101	
273	688	157	503	
161	119	110	588	
319	612	546	707	
971	912	176	525	
167	578	345	825	
247	996	480	977	
485	977	674	372	
725	876	919	567	
854	274	183	187	
685	693	198	457	
794	729	644	641	
396	831	269	901	
445	395	478	975	
878	861	624	785	
248	620	569	193	
650	819	122	104	
321	332	124	684	
751	141	993	245	
333	95	228	262	
971	974	131	222	
936	12	918	696	
460	208	703	246	
4	417	431	107	
776	802	205	320	
307	727	508	958	

## 2)

Wynik dla eksperymentu z metodą przeszukiwania sekwencyjnego :

Dla danych nie posortowanych : Wyszukiwano 100 elementów. Min = 1, Max = 197, średnio 91,23

Dla danych posortowanych : Wyszukiwano 100 elementów. Min = 1, Max = 194, średnio 100,11

Wniosek : Brak znaczącej różnicy dla obu metod.

## 3)

Wynik dla eksperymentu wyszukiwania metodą podziałów dychotomicznych :

Wyszukiwano 100 elementów. Min = 1, Max = 8, średnio 6,36

Wniosek : Zauważono znaczącą różnicę, poprawę optymalizacji działania eksperymentu w porównaniu do podpunktu drugiego. W metodzie tej maksymalna liczba dostępów do pamięci jest zdecydowanie mniejsza. Jest to metoda wyraźnie lepsza od przeszukiwania sekwencyjnego.

## 4)

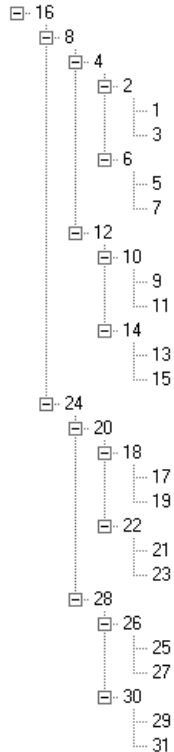
Drzewo (nie wyważone): Wyszukiwano 100 elementów. Min = 1, Max = 13, średnio 8,56

Drzewo (wyważone): Wyszukiwano 100 elementów. Min = 1, Max = 8, średnio 6,36

Wniosek: Lepsze okazały się jednak nieznacznie drzewa wyważone. Średni dostęp do danych był o krótszy oraz maksymalny potrzebny dostęp do pamięci zmniejszył się prawie dwukrotnie.

## 5)

Elementy należy wpisać w kolejności: 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. Wyważone drzewo binarne powinno wówczas wyglądać tak :



6)

Zad 6		Rozmiar strony 2	Rozmiar strony 5	Rozmiar strony 10	Rozmiar strony 50	Rozmiar strony 150	Rozmiar strony 200
Dostęp do Pamięci	Min	4	3	3	3	2	2
Dostęp do Pamięci	Max	9	9	9	8	8	8
Dostęp do Pamięci	Średnio	6,58	7,34	6,81	6,86	6,56	6,56
Dostęp do Dysku	Min	2	2	1	2	1	1
Dostęp do Dysku	Max	4	3	2	2	1	1
Dostęp do Dysku	średnio	3,56	2,93	1,94	2	1	1

Wniosek : Z obserwacji możemy wywnioskować że im większy rozmiar strony podajemy tym mniej dostępów do dysku otrzymujemy, co oznacza mniejszy czas wyszukiwania. Wówczas optymalnym ze względu na czas wyszukiwania rozmiarem strony jest 200. Rozwiązanie to jest jednak sprzeczne z ideą B-drzew.  
Przecież w B-drzewach nie chodzi o to żeby wszystkie dane znajdowałyby się w pamięci operacyjnej lecz jedynie ich część. Wówczas optymalnym rozwiązaniem staje się rozmiar strony równy 10. Zajęta wtedy pamięć operacyjna jak i ilość dostępów do dysku jest stosunkowo niska.

7)

Minimalną i maksymalną liczbę elementów jakie można umieścić w B-drzewie klasy  $t(h,m)$  :

$$N \leq (2 * m + 1)^{h-1}$$

$$N \geq 2 * (2 * m + 1)^{h-2}$$

8)

Zad 8		Rozmiar strony 2	Rozmiar strony 5	Rozmiar strony 10	Rozmiar strony 50	Rozmiar strony 150	Rozmiar strony 200
Dostęp do Pamięci	Min	5	5	4	3	2	2
Dostęp do Pamięci	Max	10	9	8	8	8	8
Dostęp do Pamięci	Średnio	7,79	6,77	6,79	6,74	6,56	6,56
Dostęp do Dysku	Min	4	3	2	2	1	1
Dostęp do Dysku	Max	4	3	2	2	1	1
Dostęp do Dysku	średnio	4	3	2	2	1	1

Na podstawie eksperymentu możemy powiedzieć że wraz ze wzrostem rozmiaru strony spada ilość dostępów do dysku. Również należy zwrócić uwagę iż liczba ta jest stała dla każdego rozmiaru strony – brak tutaj wartości minimalnej i maksymalnej. Natomiast jeśli chodzi o dostęp do pamięci, to najmniejszy jest on w przypadku rozmiaru strony 150 oraz 200. Wartości te możemy przyjąć za najbardziej optymalne. Jest to bardzo podobne do B-drzewa gdzie również zwiększanie rozmiaru strony powoduje zmniejszenie czasu wyszukiwania. Jeżeli jednak chcielibyśmy ograniczyć zajęcie pamięci operacyjnej, wówczas powinniśmy wybrać rozmiar strony równy 10, dla którego liczba dostępów do dysku i do pamięci jest stosunkowo najmniejsza.

9)

Minimalnie lepsze okazały się B-drzewa. Różnica jest jednak niewielka. Pod względem ilości dostępów do dysku, dla każdej testowanej wielkości rozmiaru strony B-drzewa okazały się bardziej wydajne. Zwróćmy jednak uwagę na to że wraz ze wzrostem rozmiaru strony różnica ta maleje. Biorąc natomiast pod uwagę ilość dostępów do pamięci, sytuacja wygląda tak samo. Tutaj również B-drzewo jest bardziej wydajne, a różnica maleje wraz ze wzrostem rozmiaru strony.

10)

Wielkość tablicy 250	Dzielenie przez rozmiar tablicy			Podział, składanie i dzielenie			Wycięcie 3cyfr klucza i normalizacja			Mieszanie Fibonacciego		
Sondowanie liniowe z krokiem 1	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia
	1	13	1,9	1	10	1,8	1	197	91,23	1	30	2,17
Sondowanie liniowe z krokiem 7	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia
	1	10	2,02	1	10	1,85	1	30	28,34	1	16	3,59
Podwójne mieszanie zależne	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia
	1	32	2,33	1	9	1,51	1	197	91,23	brak	brak	brak
Podwójne mieszanie niezależne	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia
	1	11	1,73	1	11	1,73	brak	brak	brak	1	14	2,6

Wielkość tablicy 300	Dzielenie przez rozmiar tablicy			Podział, składanie i dzielenie			Wycięcie 3cyfr klucza i normalizacja			Mieszanie Fibonacciego		
Sondowanie liniowe z krokiem 1	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia
	1	10	2,23	1	10	2,23	1	197	91,23	1	4	1,17
Sondowanie liniowe z krokiem 7	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia
	1	9	2,01	1	9	2,01	1	30	28,34	1	6	1,35
Podwójne mieszanie zależne	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia
	1	14	2,62	1	9	2,62	1	197	91,23	1	12	1,41
Podwójne mieszanie niezależne	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia
	1	6	1,53	1	6	1,53	brak	brak	brak	1	8	1,37

Wielkość tablicy 400	Dzielenie przez rozmiar tablicy			Podział, składanie i dzielenie			Wycięcie 3cyfr klucza i normalizacja			Mieszanie Fibonacciego		
Sondowanie liniowe z krokiem 1	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia
	1	9	1,72	1	9	1,72	1	197	91,23	1	8	1,37
Sondowanie liniowe z krokiem 7	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia
	1	4	1,46	1	4	1,46	1	197	91,23	1	5	1,24
Podwójne mieszanie zależne	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia
	1	7	1,66	1	7	1,66	1	197	91,23	1	4	1,22
Podwójne mieszanie niezależne	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia	Min	Max	Średnia
	1	4	1,08	1	4	1,08	brak	brak	brak	1	4	1,16

Wniosek : Dla tablicy o różnej wielkości, większej niż ilości danych zauważono :

- Dla dzielenia przez rozmiar tablicy : wartość minimum jest zawsze równa 1, wartość maksimum wraz ze wzrostem tablicy znacznie maleje, wartość średnia nie aż tak bardzo różni się w zależności od sposobu rozwiązywania kolizji.
- Dla podziału, składania i dzielenia : wartość minimum zawsze wynosi 1, wartość maksimum nieznacznie maleje bądź jest taka sama w zależności od rozmiaru tablicy, średnia nie aż tak bardzo różni się w zależności od sposobu rozwiązywania kolizji.
- Dla wycięcia 3 cyfr klucza i normalizacji : wszystkie wyniki są takie same, brak różnic, jedynie dla podwójnego mieszania niezależnego nie udało się przeprowadzić eksperymentu z powodu braku miejsca o którym poinformował nas program. Sprawdzono że dla danych w ilości danych ze sprawozdania przy wielkości tablicy równej 500 dopiero program pozwoli na wykonanie eksperymentu.
- Dla mieszania Fibonacciego : wartość minimum jest zawsze równa 1, wartość maksimum wraz ze wzrostem tablicy znacząco maleje wraz ze wzrostem wielkości tablicy, wartość średnia widocznie maleje wraz ze wzrostem tablicy. Nie udało się przeprowadzić eksperymentu dla wielkości tablicy 250 i sposobu rozwiązywania kolizji podwójne mieszanie zależne.

Na podstawie przeprowadzonego eksperymentu stwierdzam że najbardziej optymalnym wynikiem było zastosowanie tablicy o rozmiarze 400 i większym dla „dzielenia przez rozmiar tablicy i „podziału, składania i dzielenia” wraz ze sposobem rozwiązywania kolizji Podwójne mieszanie niezależne oraz dla metody mieszanie Fibonacciego ze sposobem rozwiązywania kolizji Podwójne mieszanie zależne i niezależne. Zwiększenie rozmiaru tablicy znacząco zmniejsza liczbę występujących kolizji jednak prowadzi do znaczącego marnotrawstwa zasobów. Przy bardzo dużych rozmiarach tablic liczba kolizji przestaje spadać i osiąga stały poziom równy 1, natomiast tak jak napisałem wcześniej im większa tablica tym więcej zasobów marnujemy.