STRUKTURY BAZ DANYCH – PROJEKT NR 1

Damian Strojek, s184407

1 Wprowadzenie

Celem projektu było zaimplementowanie algorytmu scalania naturalnego służącego do sortowania plików sekwencyjnych. W moim przypadku, pojedynczy rekord pliku to para liczb definiujących napięcie oraz natężenie prądu. Rekordy są uporządkowane według mocy elektrycznej.

Zadanie zrealizowałem wykorzystując metodę niewyważonego, trójtaśmowego sortowania przez scalanie naturalne. Sortowanie przez łączenie naturalne jest zalicznae do grupy sortowań zewnętrznych, tzn. operuje na taśmach. Taśmą można nazwać dowolny plik o dostępie sekwencyjnym, niemożliwe jest więc wykorzystywanie tradycyjnych metod sortowania. Słowo „seria", używane w tym dokumencie, rozumiane jest jako najdłuższy posortowany podciąg rekordów.

Algorytm rozpoczyna się od kopiowana danych z pliku źródłowego (input.csv) do taśmy c (tape_c.csv) – głównym powodem jest bezpieczeństwo oryginalnych danych. Następnie przechodzimy do fazy dzielenia rekordów – w taśmie c szukamy serii i kopiujemy je na przemian do taśmy a (tape_a.csv) oraz taśmy b (tape_b.csv). Po tej operacji w taśmach powinna się znajdować równa liczba serii. Wyjątkiem jest sytuacja, w której w pliku źródłowym liczba rekordów jest nieparzysta. Wtedy w taśmie a będzie znajdował się o jeden rekord więcej niż w taśmie b. Po podziale rekordów rozpoczynmay fazę łączenia, która polega na pobieraniu po jednej serii z kazdej z taśm i scalaniu ich do taśmy a. Powtarzamy ten punkt tak długo, aż taśma b będzie pusta. Oznacza to, że wszystkie dane są już posortowane i znajdują się na taśmie a. Następnie przechodzimy do kopiowania zawartości taśmy a do pliku wynikowego (sorted.csv).

2 Specyfikacje projektu

Każdy wiersz w pliku wejściowym, operacyjnym (taśmy) lub wyjściowym to osobny rekord składający się z pary napięcia i natężenia prądu. W trakcie działania programu pliki operacyjne posiadają również trzecią kolumnę definiującą moc elektryczną.

Wcześniej wspomniane pliki posiadają format `.csv`, ale bez problemu można pracować na czystych plikach tekstowych.

3 Eksperymenty

Eksperymenty zostały przeprowadzone na jedenastu plikach. Każdy z plików posiadał inną liczbę rekordów – rozpocząłem od 100 rekordów, a następnie zwiększałem ilość rekordów dwukrotnie, w stosunku do poprzedniego przebiegu. Testy skończyłem na liczbie 51200 wygenerowanych rekordów ze względu na czas operacji, który rósł wprost proporcjonalnie do ilości rekordów w pliku. We wszystkich testach blocking factor (`BUFFER_SIZE`) jest równy 100 – jest to maksymalna liczba rekordów, która może się zmieścić w bloku.

Rekordy do eksperymentów generowane są losowo za pomocą funckji `generateRandomRecords()`. Każdy rekord jest sumą dwóch, losowo wygenerowanych liczb – części całkowitej oraz części ułamkowej napięcia lub natężenia prądu.

Wszystkie testy zostały przeprowadzone trzykrotnie, a średnią wyników otrzymanych z kolejnych testów przedstawiam poniżej.

4 Wyniki liczbowe eksperymentów

L.p.	Liczba rekordów w pliku źródłowym	Liczba odczytów z dysku	Liczba zapisów na dysk	Łączna liczba operacji na dysku	Liczba przebiegów (rozdzielenie i łączenie)
1	100	32 (54%)	27 (46%)	59	7
2	200	53 (52%)	48 (48%)	101	8
3	400	97 (51%)	93 (49%)	190	9
4	800	191 (50%)	190 (50%)	381	10
5	1600	393 (50%)	399 (50%)	792	11
6	3200	827 (49%)	848 (51%)	1675	12
7	6400	1757 (49%)	1809 (51%)	3566	13
8	12800	3743 (49%)	3858 (51%)	7601	14
9	25600	7969 (49%)	8211 (51%)	16180	15
10	51200	16931 (49%)	17428 (51%)	34359	16
SUMA	**-**	**31993 (49%)**	**32911 (51%)**	**64904**	**-**

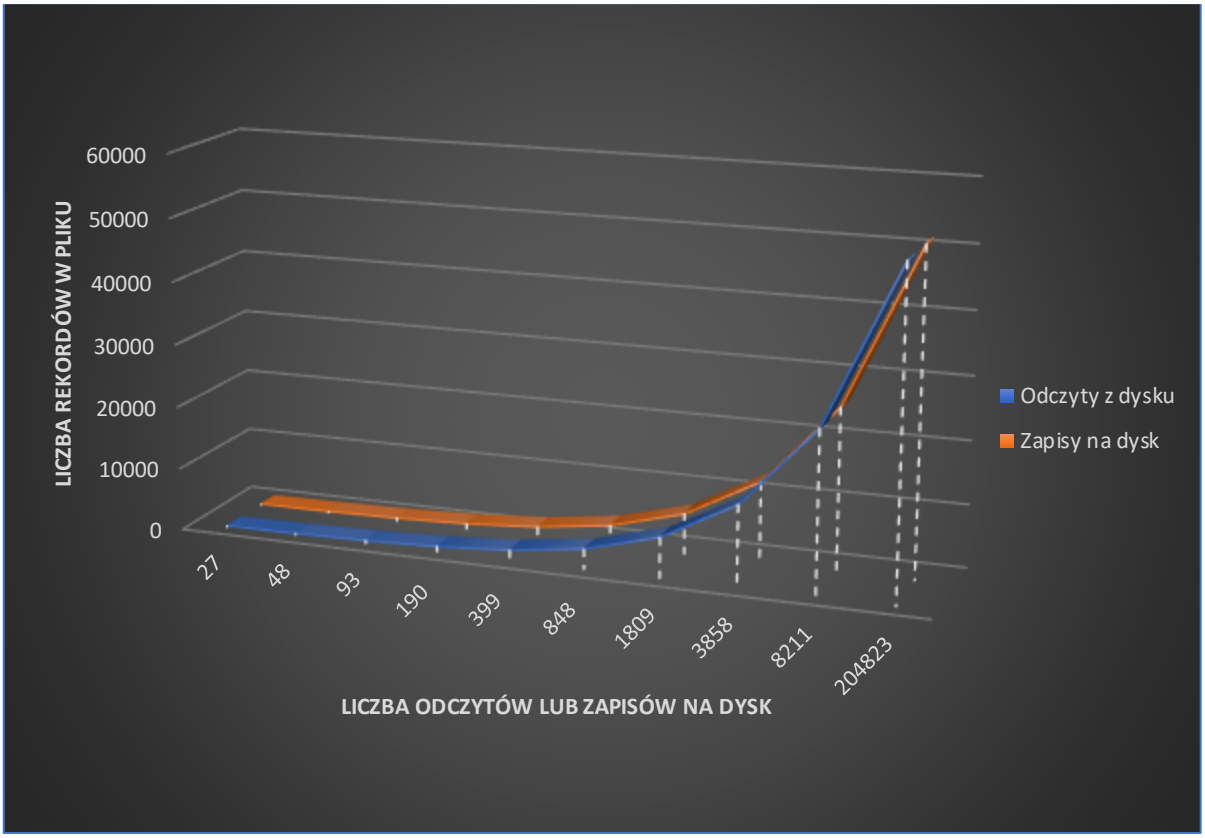
5 Wyniki teoretyczne (spodziewane) eksperymentów

Dla maksymalnych wartości przyjąłem, że $r = N$, a dla wartości średnich, że $r = \frac{N}{2}$. W przypadku sortowania przez scalanie naturalne, jeżeli mamy r serii wykonamy maksymalnie $\lceil \log_2 N \rceil$ przebiegów. Liczba ta może być mniejsza ze względu na łączenie się serii w pliku. Dodatkowo, łączna liczba operacji odczytu z dysku oraz zapisu na dysk nie powinna przekroczyć $4 * N * \lceil \log_2 r \rceil * \frac{1}{b}$.

L.p.	Liczba rekordów w pliku źródłowym	Teoretyczna, maksymalna sumaryczna liczba operacji na dysku	Teoretyczna maksymalna liczba przebiegów
1	100	28	7
2	200	64	8
3	400	144	9

4	800	320	10
5	1600	704	11
6	3200	1536	12
7	6400	3328	13
8	12800	7168	14
9	25600	15360	15
10	51200	32768	16
SUMA	**-**	**61420**	**-**

6 Wykres liczby odczytów z dysku i zapisów na dysk w zależności od liczby rekordów w pliku źródłowym



7 Wnioski z eksperymentów

Pierwsza obserwacja dotyczy liczby odczytów z dysku, która jest większa niż liczba zapisów na dysk do momentu przekroczenia 1600 rekordów w pliku źródłowym. Od tego momentu program częściej zapisuje na dysk niż odczytuje z niego dane.

Liczba przebiegów rośnie bardzo powoli i jest przewidywalna. Wartości teoretyczne odbiegają od wartości, które zostały policzone przez mój program. Wynikać to może ze specyfiki języka lub mniej optymalnego podejścia algorytmicznego z mojej strony.