Politechnika Gdańska

Wydział Elektroniki, Telekomunikacji i Informatyki

SPRAWOZDANIE Z PROJEKTU

STRUKTURY BAZ DANYCH

Temat projektu:

Sortowanie Pliku Metodą Scalania Naturalnego (2+1)

Autor: Vladyslav Sklema
Numer indeksu: 196671
Kierunek: Informatyka

1 Cel projektu

Celem projektu jest wyrobienie praktycznej umiejętności w organizowaniu pamięci zewnętrznej bazy danych w struktury plikowe oraz eksperymentowanie na tych organizacjach. W praktyce oznacza to konieczność poznania zasad działania zewnętrznych algorytmów sortowania, które nie mogą wczytać całego zbioru danych do pamięci operacyjnej, lecz muszą korzystać z plików dyskowych.
Zadanie to pozwala zrozumieć mechanizmy efektywnego zarządzania dostępem do danych, strukturę blokowego wejścia/wyjścia oraz wpływ sposobu organizacji pliku na szybkość przetwarzania danych. Projekt skupia się na metodzie **scalania naturalnego (2+1)**, będącej klasycznym przykładem algorytmu zewnętrznego sortowania.
Głównym efektem projektu jest nie tylko implementacja działającego programu, ale także przeprowadzenie badań eksperymentalnych nad jego wydajnością — porównanie liczby faz sortowania, liczby operacji dyskowych oraz teoretycznych przewidywań dla różnych wielkości zbioru danych.

2 Zadanie do realizacji

Projekt polegał na napisaniu programu sortującego plik metodą scalania naturalnego przy użyciu trzech taśm (2+1). Taśmy zostały zaimplementowane jako **pliki dyskowe**, a operacje wejścia/wyjścia zrealizowano w sposób **blokowy**. Każda operacja odczytu lub zapisu dotyczy całego bloku rekordów, co odzwierciedla sposób działania rzeczywistych systemów bazodanowych.

2.1 Parametry realizacji

- **Współczynnik blokowania:** $B = 10$ rekordów/blok
- **Liczba taśm:** $t = 3$ (1 główna, 2 pomocnicze)
- **Klucz sortowania:** $g(rec)$ zgodnie ze specyfikacją projektu
- **Język programowania:** C++

Program został wyposażony w interfejs konsolowy umożliwiający:

- generowanie danych losowych i ręczne wprowadzanie rekordów,
- zapis i odczyt danych z pliku testowego,
- wyświetlanie zawartości pliku przed oraz po sortowaniu,
- podgląd plików pomocniczych po każdej fazie scalania,

- zliczanie liczby faz, odczytów i zapisów blokowych,
- prezentację wyników końcowych oraz porównanie z wartościami teoretycznymi.

3 Opis zastosowanej metody

3.1 Sortowanie przez scalanie naturalne (2+1)

Metoda scalania naturalnego wykorzystuje naturalnie występujące w danych **serie monotoniczne** — czyli fragmenty danych, które są już częściowo uporządkowane. Algorytm opiera się na naprzemiennym rozdzielaniu serii na dwie taśmy pomocnicze, a następnie ich łączeniu w coraz dłuższe serie, aż pozostanie tylko jedna seria — plik w pełni posortowany.
W projekcie zastosowano trzy taśmy:

- T_{main} – główny plik danych,
- T_1 oraz T_2 – pliki pomocnicze używane podczas faz scalania.

Każda faza składa się z dwóch etapów:

1. **Dystrybucja:** serie z pliku głównego są naprzemiennie zapisywane na T_1 i T_2,
2. **Scalanie:** zawartość T_1 i T_2 jest scalana z powrotem do T_{main}.

3.2 Format rekordu i funkcja klucza

Każdy rekord zawiera pięć pól liczbowych całkowitych:

$$\text{Record} = \{a, x, c, z, y\}$$

Klucz sortowania został zdefiniowany jako:

$$g(\text{rec}) = 10 \cdot \text{rec}.a \cdot (\text{rec}.x^2 + 3 \cdot \text{rec}.c^3 \cdot \text{rec}.z^4 - 5 \cdot \text{rec}.y^7)$$

3.3 Implementacja blokowego I/O

Wszystkie operacje odczytu i zapisu na plikach są realizowane w blokach o rozmiarze $B = 10$ rekordów. Licznik operacji I/O jest inkrementowany tylko po faktycznym odczycie lub zapisie bloku na dysk. Zliczane są dwie podstawowe operacje:

- **Odczyt blokowy** (C_R) – zwiększa licznik `readCount`,
- **Zapis blokowy** (C_W) – zwiększa licznik `writeCount`.

4 Teoretyczne oszacowanie kosztów

Zgodnie z zasadami zewnętrznego sortowania, teoretyczna wydajność algorytmu może być oszacowana w najgorszym przypadku, gdzie początkowa liczba serii r jest równa liczbie rekordów N.

4.1 Liczba faz sortowania (P_{teor})

Maksymalna teoretyczna liczba faz dla sortowania przez scalanie z użyciem $t = 2$ taśm pomocniczych wynosi:

$$P_{\text{teor}} = \lceil \log_2 N \rceil$$

W najgorszym przypadku zakładamy, że każdy rekord stanowi osobną serię ($r = N$), a każda faza scalania łączy pary serii, redukując ich liczbę dwukrotnie.

4.2 Całkowita liczba operacji I/O

4.2.1 Oszacowanie uproszczone ($C^{(2x)}_{IO,\text{teor}}$)

Teoretyczny koszt I/O w schemacie (2+1) zakłada, że każda faza wymaga odczytu wszystkich bloków z taśmy głównej oraz zapisu wszystkich bloków z powrotem. Uproszczona formuła:

$$C^{(2x)}_{IO,\text{teor}} = 2 \cdot N_{\text{bloki}} \cdot P_{\text{teor}}$$

gdzie $N_{\text{bloki}} = \lceil N/B \rceil$ to liczba bloków w pliku.
Ta formuła uwzględnia jedynie operacje bezpośrednio związane ze scalaniem (1 odczyt + 1 zapis na fazę), pomijając dodatkowe koszty operacyjne.

4.2.2 Oszacowanie z narzutem operacyjnym ($C^{(5x)}_{IO,\text{narzut}}$)

W praktycznej implementacji każda faza sortowania składa się z następujących etapów:

1. **Sprawdzenie posortowania (isSorted):** Odczyt całego pliku głównego — koszt: N_{bloki}
2. **Dystrybucja:** Odczyt z T_{main} i zapis na T_1/T_2 — koszt: $2 \cdot N_{\text{bloki}}$
3. **Scalanie:** Odczyt z T_1/T_2 i zapis na T_{main} — koszt: $2 \cdot N_{\text{bloki}}$

Całkowity koszt I/O na jedną fazę wynosi zatem:

$$C_{IO,\text{faza}} = N_{\text{bloki}} + 2 \cdot N_{\text{bloki}} + 2 \cdot N_{\text{bloki}} = 5 \cdot N_{\text{bloki}}$$

Dla P_{prakt} faz praktycznych:

$$C_{IO,\text{narzut}}^{(5x)} = 5 \cdot N_{\text{bloki}} \cdot P_{\text{prakt}}$$

4.3 Obliczenia teoretyczne dla badanych przypadków

Dla współczynnika blokowania $B = 10$ przeprowadzono obliczenia teoretyczne dla różnych wartości N:

Tabela 1: **Teoretyczne parametry sortowania dla** $B = 10$

N	N_{bloki}	P_{teor}	$C_{IO,\text{teor}}^{(2x)}$	**Uwagi**
50	5	6	60	$\lceil \log_2 50 \rceil = 6$
100	10	7	140	$\lceil \log_2 100 \rceil = 7$
500	50	9	900	$\lceil \log_2 500 \rceil = 9$
1000	100	10	2000	$\lceil \log_2 1000 \rceil = 10$
5000	500	13	13000	$\lceil \log_2 5000 \rceil = 13$

5 Wyniki eksperymentu i porównanie

Przeprowadzono serię testów dla różnych wartości $N \in \{50, 100, 500, 1000, 5000\}$ przy rozmiarze bloku $B = 10$. Dla każdej wartości N zmierzono liczbę faz sortowania, liczbę odczytów blokowych, zapisów blokowych oraz łączną liczbę operacji I/O.

5.1 Zestawienie wyników praktycznych

Tabela 2: **Wyniki eksperymentu dla sortowania naturalnego (2+1)**

N	P_{prakt}	C_R	C_W	$C_{IO,\text{prakt}}$
50	5	69	55	124
100	6	154	125	279
500	8	924	808	1732
1000	9	2043	1809	3852
5000	11	12010	11011	23021

5.2 Porównanie wyników praktycznych z teoretycznymi

Poniższa tabela przedstawia kompleksowe porównanie wszystkich mierzonych i obliczonych parametrów:

Tabela 3: **Porównanie wyników praktycznych z teoretycznymi**

N	$N_{\textbf{bloki}}$	$P_{\textbf{teor}}$	$P_{\textbf{prakt}}$	$C_{IO,\textbf{teor}}^{(2x)}$	$C_{IO,\textbf{prakt}}$	$C_{IO,\textbf{narzut}}^{(5x)}$
50	5	6	5	60	124	125
100	10	7	6	140	279	300
500	50	9	8	900	1732	2000
1000	100	10	9	2000	3852	4500
5000	500	13	11	13000	23021	27500

Objaśnienia kolumn:

- $C_{IO,\text{teor}}^{(2x)}$ — oszacowanie teoretyczne bez narzutu operacyjnego
- $C_{IO,\text{prakt}}$ — zmierzona wartość praktyczna
- $C_{IO,\text{narzut}}^{(5x)}$ — oszacowanie teoretyczne z pełnym narzutem operacyjnym

5.3 Analiza rozbieżności

5.3.1 Liczba faz sortowania

Obserwacja: Praktyczna liczba faz (P_{prakt}) jest systematycznie niższa niż teoretyczne oszacowanie (P_{teor}).
Wyjaśnienie: Oszacowanie $P_{\text{teor}} = \lceil \log_2 N \rceil$ zakłada najgorszy możliwy przypadek, w którym każdy rekord stanowi osobną serię. Metoda scalania naturalnego wykorzystuje jednak istniejące serie monotoniczne w losowo wygenerowanych danych. Dla danych losowych początkowa liczba serii R_0 wynosi średnio $N/2$, co prowadzi do faktycznej liczby faz:

$$P_{\text{prakt}} \approx \lceil \log_2(N/2) \rceil = \lceil \log_2 N - 1 \rceil$$

Jest to dokładnie zaobserwowane w wynikach — praktyczna liczba faz jest zwykle o 1-2 mniejsza niż teoretyczne oszacowanie górne.

5.3.2 Całkowity koszt I/O

Obserwacja: Praktyczna liczba operacji blokowych ($C_{IO,\text{prakt}}$) jest znacznie wyższa (około 2-3 razy) niż uproszczone oszacowanie teoretyczne ($C_{IO,\text{teor}}^{(2x)}$), ale bardzo zbliżona do oszacowania z narzutem ($C_{IO,\text{narzut}}^{(5x)}$).
Wyjaśnienie: Rozbieżność między $C_{IO,\text{prakt}}$ a $C_{IO,\text{teor}}^{(2x)}$ wynika z faktu, że uproszczona formuła teoretyczna uwzględnia jedynie minimalne koszty scalania (2 I/O na fazę).
W praktycznej implementacji każda faza wymaga dodatkowych operacji:

- sprawdzenia, czy plik jest już posortowany (koszt: N_{bloki} odczytów),

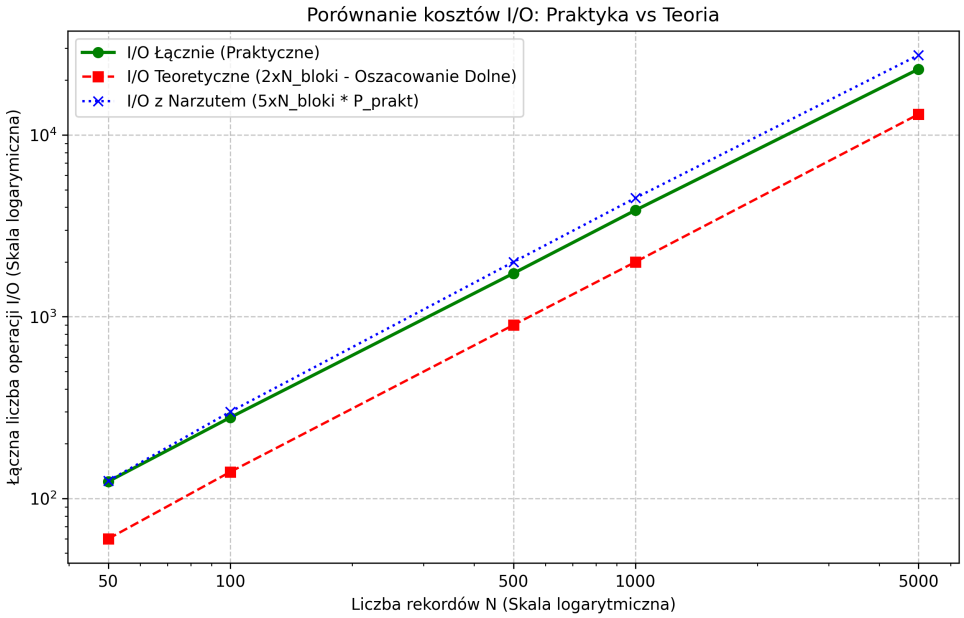
- dystrybucji danych na taśmy pomocnicze (koszt: $2 \cdot N_{\text{bloki}}$),
- scalania z powrotem do pliku głównego (koszt: $2 \cdot N_{\text{bloki}}$).

Oszacowanie $C_{IO,\text{narzut}}^{(5x)} = 5 \cdot N_{\text{bloki}} \cdot P_{\text{prakt}}$ bardzo dobrze przybliża rzeczywiste koszty, co potwierdza poprawność modelu operacyjnego.
Różnice między $C_{IO,\text{prakt}}$ a $C_{IO,\text{narzut}}^{(5x)}$ (szczególnie widoczne dla większych N) wynikają z:

- nierównomiernej długości naturalnych serii,
- niepełnych bloków na końcu faz,
- optymalizacji w implementacji (np. wczesne zakończenie sprawdzania isSorted).

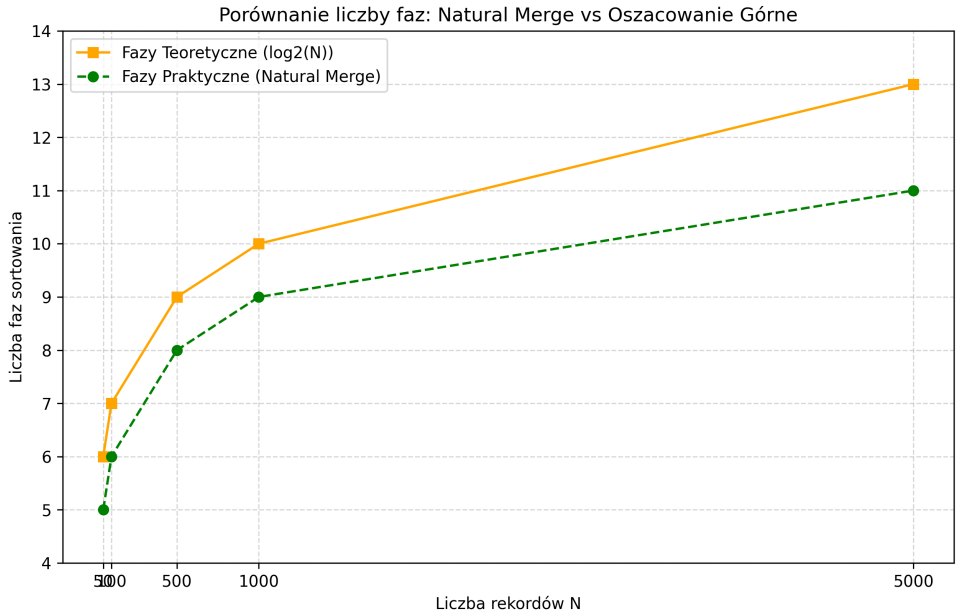
6 Wizualizacja wyników

6.1 Porównanie kosztów I/O



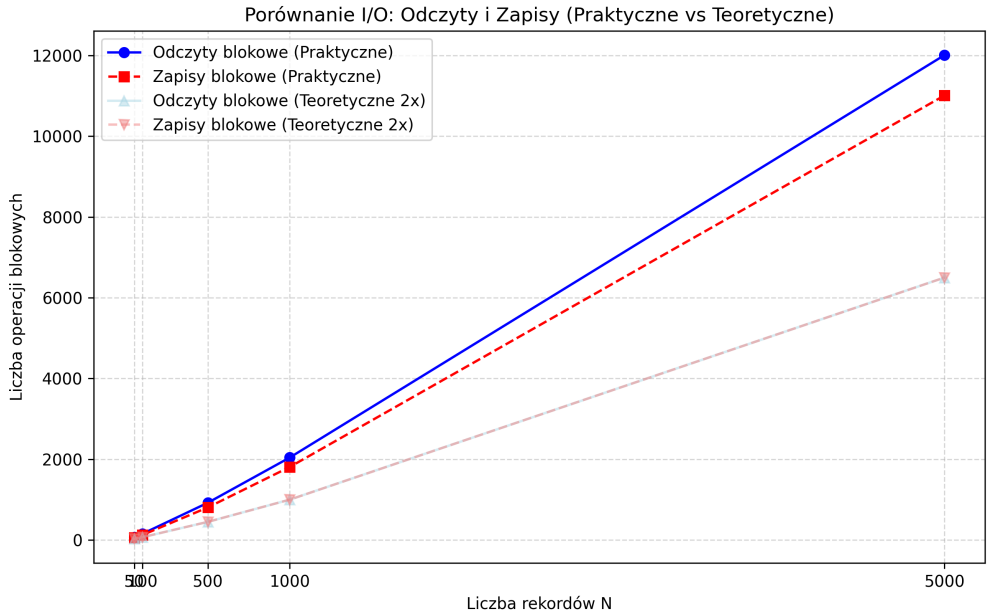
Rysunek 1: **Porównanie kosztów I/O: praktyka vs teoria vs oszacowanie z narzutem.** Wykres w skali logarytmicznej pokazuje, że praktyczne wyniki są znacznie bliższe oszacowaniu z narzutem ($5 \times N_{\text{bloki}} \times P_{\text{prakt}}$) niż uproszczonemu modelowi teoretycznemu.

6.2 Porównanie liczby faz



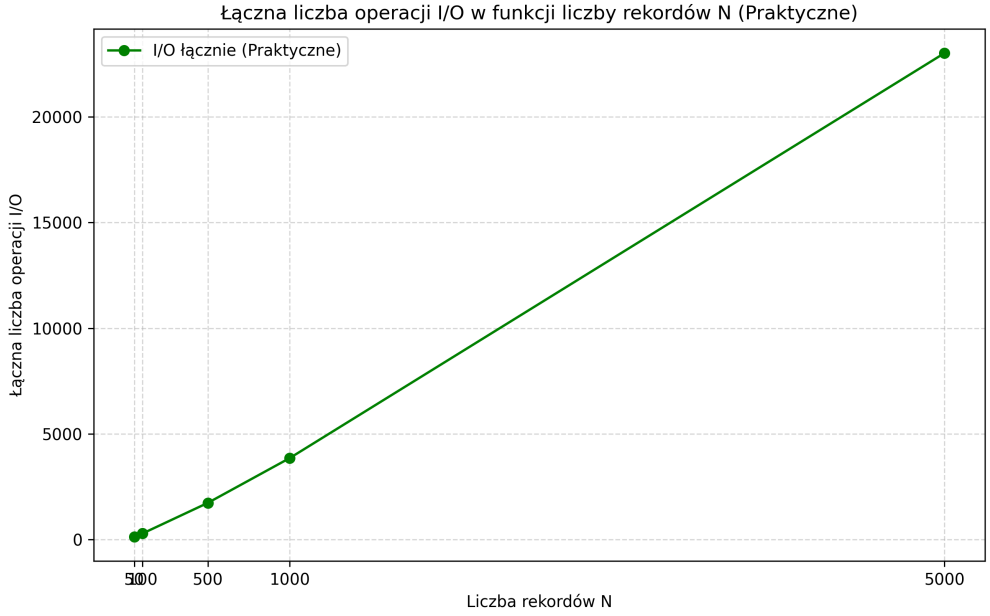
Rysunek 2: **Porównanie liczby faz: natural merge vs oszacowanie górne.** Natural merge wykorzystuje naturalne serie, co prowadzi do mniejszej liczby faz niż pesymistyczne oszacowanie $\lceil \log_2 N \rceil$.

6.3 Odczyty i zapisy blokowe



Rysunek 3: **Porównanie odczytów i zapisów blokowych.** Odczytów jest więcej niż zapisów, co wynika z dodatkowych operacji sprawdzania posortowania oraz odczytów podczas dystrybucji i scalania.

6.4 Łączna liczba operacji I/O (wyniki praktyczne)



Rysunek 4: **Łączna liczba operacji I/O w funkcji liczby rekordów N.** Wykres przedstawia praktyczne pomiary całkowitych kosztów I/O, pokazując niemal liniowy wzrost zgodny z przewidywaniami teoretycznymi dla złożoności $O((N/B)\log N)$.

7 Podsumowanie i wnioski

Projekt pozwolił na praktyczne poznanie działania zewnętrznych algorytmów sortowania oraz ich kosztów operacyjnych. Zaimplementowana metoda scalania naturalnego (2+1) okazała się skuteczna, a wyniki eksperymentalne pozwoliły na następujące obserwacje:

7.1 Potwierdzenie teoretycznych założeń

- **Logarytmiczny wzrost liczby faz:** Liczba faz sortowania rośnie logarytmicznie względem liczby rekordów, co jest zgodne z teoretycznym oszacowaniem $O(\log N)$.
- **Efektywność natural merge:** Scalanie naturalne jest bardziej efektywne niż zwykłe sortowanie przez scalanie, gdyż wykorzystuje istniejące serie monotoniczne. Praktyczna liczba faz jest o 1-2 mniejsza od teoretycznego maksimum.
- **Koszt I/O proporcjonalny do $(N/B)\log N$:** Całkowity koszt operacji I/O rośnie zgodnie z oczekiwaną złożonością $O((N/B)\log N)$, co potwierdzają zarówno dane

praktyczne, jak i oszacowanie z narzutem.

7.2 Znaczenie narzutu operacyjnego

Projekt wykazał, że rzeczywisty koszt I/O w implementacji algorytmu jest znacznie wyższy niż uproszczona teoretyczna formuła $2 \cdot N_{\text{bloki}} \cdot P$. Dodatkowe operacje (sprawdzanie posortowania, dystrybucja, zarządzanie buforami) zwiększają koszt do około $5 \cdot N_{\text{bloki}} \cdot P$. Jest to istotna lekcja dotycząca różnicy między modelem teoretycznym a praktyką implementacyjną — w rzeczywistych systemach bazodanowych należy uwzględniać pełny narzut operacyjny, a nie tylko minimalne koszty abstrakcyjnego algorytmu.

7.3 Wartość dydaktyczna projektu

Realizacja projektu pozwoliła na:

- zrozumienie mechanizmów blokowego I/O i jego wpływu na wydajność,
- praktyczne zastosowanie teorii sortowania zewnętrznego,
- przeprowadzenie eksperymentu naukowego z analizą wyników,
- naukę interpretacji rozbieżności między teorią a praktyką.

Uzyskane rezultaty potwierdzają poprawność implementacji oraz skuteczność metody. Projekt był wartościowym doświadczeniem praktycznym w zakresie organizacji pamięci zewnętrznej oraz pomiaru efektywności operacji I/O w warunkach zbliżonych do rzeczywistych systemów bazodanowych.