

# Mnożenie macierzy

## Algorytm Cannon'a

### 1. Wstęp teoretyczny

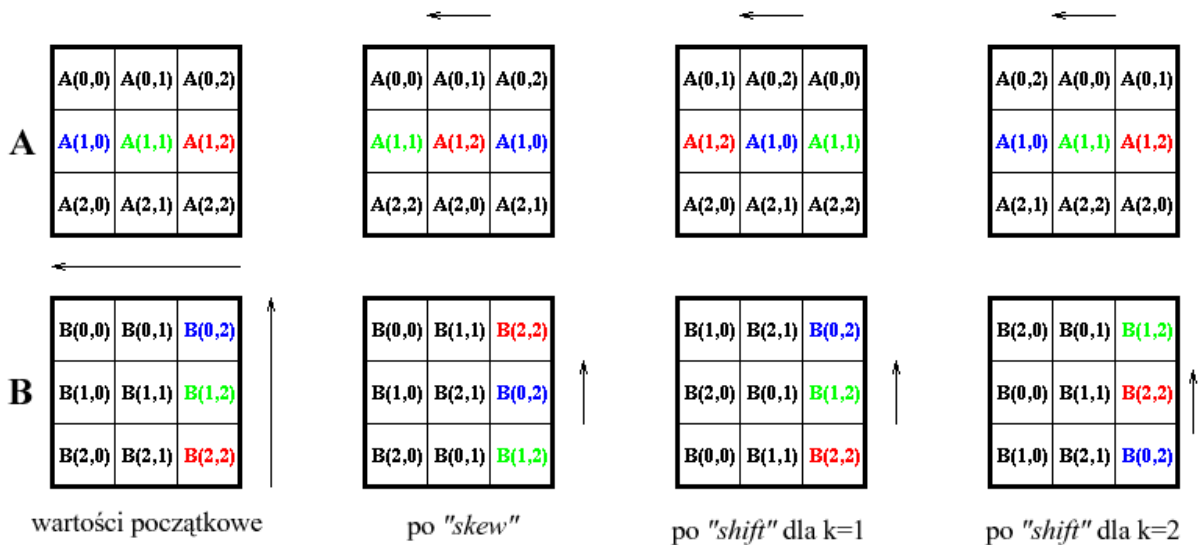
Algorytm Cannon'a jest rozproszonym algorytmem mnożenia macierzy dla siatek dwuwymiarowych, który po raz pierwszy opisany został w 1969 r. przez Lynn'a Elliot'a Cannon'a.

- Algorytm jest efektywny jedynie dla struktury kwadratowej,
- Działających w algorytmie procesorów jest tyle jaki jest rozmiar macierzy

#### Algorytm Cannon'a:

1. Oznaczamy mnożone macierze jako **A** i **B**, macierz wynikową jako **C**, macierz procesów jako **P**.
2. Proces **P(i, j)** początkowo przechowuje **A(i, j)**, a **B(i, j)** oblicza blok **C(i, j)** macierzy wynikowej.
3. Przekształcamy **A** i **B** w taki sposób, aby każdy proces mógł niezależnie rozpocząć **mnożenie** swoich lokalnych podmacierzy.  
Przesuwamy wszystkie podmacierze **A(i, j)** w lewo o **i** kroków i wszystkich podmacierzy **B(i, j)** w górę o **j** kroków.
4. Wykonujemy mnożenie bloków lokalnych.
5. Każdy blok **A** przesuwamy o jeden krok w lewo, a każdy blok **B** przesuwamy o jeden krok w górę.
6. Wykonujemy mnożenie kolejnych bloków, dodajemy do wyniku częściowego i powtarzamy to, aż **wszystkie bloki zostaną pomnożone**.

## Przykład obliczania jednego z elementów macierzy wynikowej



$$C(1, 2) = A(1, 0) * B(0, 2) + A(1, 1) * B(1, 2) + A(1, 2) * B(2, 2)$$

## 2. Implementacja

Program został napisany w języku C.

Korzystamy z jednej struktury danych o nazwie *matrix\_data*.

Struktura składa się z:

- *mat* - spłaszczonej, jednowymiarowej tablicy elementów macierzy w postaci zmiennoprzecinkowej,
- *row* - ilość rzędów macierzy,
- *col* - ilość kolumn macierzy

W programie możemy znaleźć 4 funkcje:

- *main* - główna funkcja zajmująca się przebiegiem algorytmu, czyli obliczeniami i przekształceniami,
- *print\_matrix* - funkcja wypisująca macierz do konsoli w sposób sformatowany,

- ***initialize\_matrix*** - funkcja inicjalizująca macierz poprzez wczytanie jej z pliku **.csv**, zaalokowanie pamięci i zapisanie w postaci struktury *matrix\_data*,
- ***save\_matrix*** - funkcja zapisująca macierz w postaci pliku **.csv**, używają

Program obsługuje flagę -v (verbose). Po uruchomieniu programu z tą flagą możemy zobaczyć wypisane w sposób **macierze mnożone** oraz **macierz wynikową**.

### 3. Uruchomienie

Do przygotowania i uruchomienia rozwiązania służy plik **makefile**, w którym zdefiniowane są następujące komendy:

- **build** - komenda kompiluje nasz program,
- **build\_valgrind** - j.w korzystając z valgrinda,
- **nodes** - tworzy plik z węzłami,
- **run** - uruchamia program korzystając z wielu komputerów,
- **run\_one** - uruchamia program na jednym komputerze,
- **run\_one\_verb** - jw. z parametrem verbose,
- **run\_one\_valgrind** - jw. korzystając z valgrinda,
- **clean** - usuwa utworzone przy uruchamianiu pliki.

### 4. Zawartość katalogu

- **main.c** - program obsługujący zadanie,
- **makefile** - plik make pozwalający na łatwe uruchamianie,
- **A.csv** - pierwsza mnożona macierz,
- **B.csv** - druga mnożona macierz,
- **Result.csv** - przykład docelowej wynikowej macierzy,
- **Dokumentacja.pdf** - opis i dokumentacja projektu.

### 5. Materiały

- Prezentacja - [LINK](#)
- GitHub - [LINK](#)