

Testowanie Iloczynu Dwóch Macierzy

Weronika Orzechowska

1 Zadanie

Napisz program, który dla danych na wejściu trzech kwadratowych macierzy A , B , oraz C rozmiaru $n \times n$, sprawdzi, czy C jest iloczynem macierzy A oraz B (tzn. czy $A \cdot B = C$). Zaproponuj algorytm, który działa w czasie $O(n^2)$. Porównaj jego czas działania z algorytmem $O(n^3)$ wymagającym przemnożenia macierzy A oraz B .

2 Implementacja

2.1 Algorytm $O(n^2)$

Algorytm $O(n^2)$ to algorytm probabilistyczny działający na podstawie mnożenia macierzy przez wektor losowy.

Pierwszym krokiem było losowe wybranie wartości p z dziesięcio-elementowej listy zawierającej jedenastocyfrowe liczby pierwsze. Kolejne obliczenia będziemy wykonywać *mod* p . Na podstawie wartości p tworzymy losowy wektor x wymiaru $n \times 1$

Następnie wykonujemy obliczenie:

$$m = B \cdot x \pmod{p}$$

w kolejnym kroku:

$$n = A \cdot m \pmod{p}.$$

Wyznaczenie tego wektora n to operacja o złożoności $O(n^2)$.

Drugą stronę równości traktujemy analogicznie, wyznaczając wektor k :

$$n = C \cdot x \pmod{p}.$$

Istotnie, jeśli $A \cdot B = C$, to zachodzi równość wektorów n i k . Ponieważ wektor x jest losowany dla bardzo dużych liczb pierwszych, to z dużym prawdopodobieństwem, jeśli $A \cdot B \neq C$, to $n \neq k$.

Sprawdzenie równości wektorów odbywa się w czasie $O(n)$.

Przy wywoływaniu tego algorytmu istotnie otrzymujemy poprawne wyniki, zatem jego poprawność jest empirycznie potwierdzona.

2.2 Algorytm $O(n^3)$

Algorytm $O(n^3)$ to klasyczny algorytm mnożenia macierzy zgodnie ze wzorem (dla macierzy o tych samych wymiarach):

$$c_{i,j} = \sum_{r=1}^n a_{i,r} b_{r,j}$$

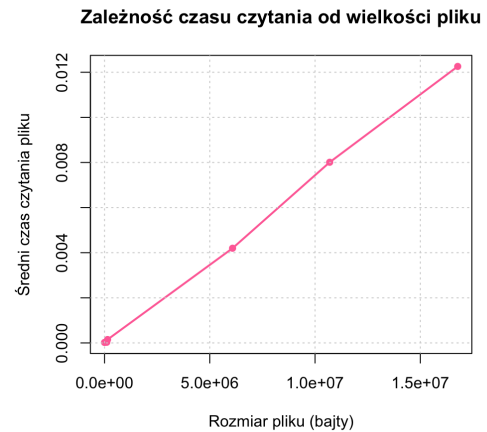
Wyznaczenie macierzy $A \cdot B$ odbywa się w czasie $O(n^3)$, natomiast sprawdzenie, czy $A \cdot B = C$ w czasie $O(n^2)$.

3 Testy

3.1 Wczytanie plików

Analiza długości wczytywania plików (jak zresztą mogliśmy się spodziewać) pokazała zależność liniową.

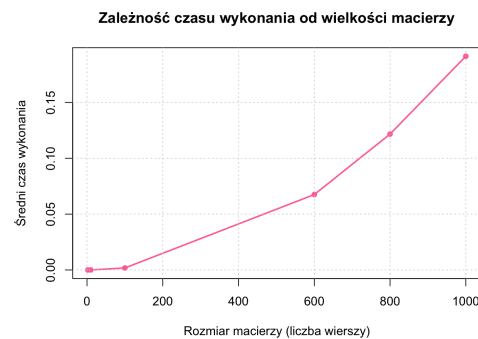
plik	średni czas wczytania (s)
plik 1	0.00002193
plik 2	0.00002997
plik 3	0.00015109
plik 4	0.00419873
plik 5	0.00800995
plik 6	0.01225807



3.2 Algorytm $O(n^2)$

Wykres pokazuje umiarkowany wzrost czasu wykonania wraz z rozmiarem macierzy. Dla większych macierzy czas rośnie, ale nadal pozostaje stosunkowo niski. Wzrost nie jest eksplodujący, co sugeruje, że czas działania jest kwadratowy.

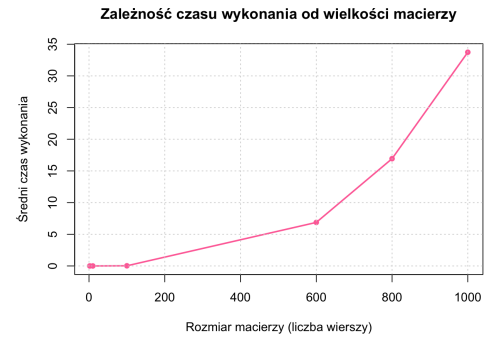
plik	średni czas wykonania (s)
plik 1	0.000004080
plik 2	0.000025558
plik 3	0.001794140
plik 4	0.067554270
plik 5	0.121732600
plik 6	0.191460955



3.3 Algorytm $O(n^3)$

Widzimy, że wykres zależności czasu wykonania od rozmiaru macierzy, wskazuje wyraźnie rosnący trend, co jest zgodne z oczekiwanym zachowaniem algorytmu o złożoności $O(n^3)$. Dla małych macierzy, czas wykonania jest bliski zeru, ale przy większych rośnie w sposób wykładniczy. W porównaniu z $O(n^2)$ wzrost jest znacznie bardziej stromy.

plik	średni czas wykonania (s)
plik 1	0.000002050
plik 2	0.000037342
plik 3	0.026288660
plik 4	6.873634613
plik 5	16.932529495
plik 6	33.733413740



3.4 Porównanie

Najbardziej oczywistą obserwacją jest gwałtownie rosnący czas wykonania dla algorytmu $O(n^3)$ w porównaniu do algorytmu $O(n^2)$. Wzrost czasu dla klasycznej metody mnożenia macierzy jest bardzo szybki – dla dużych macierzy czas sięga już kilkudziesięciu sekund, podczas gdy metoda probabilistyczna nadal pozostaje praktycznie niezauważalna na skali wykresu.

