Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних інформаційних систем

Алгоритми та складність

Завдання №5 “ Алгоритм Штрассена для множення матриць”

Виконав студент 2-го курсу

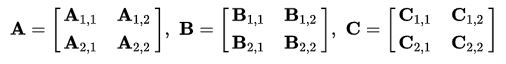
Групи К-29

Короткий Ростислав Сергійович

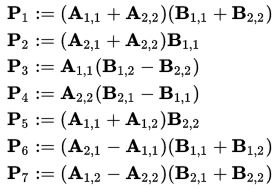
Аналіз алгоритму

Алгоритм Штрассена:

Нехай A і B – дві n\*n матриці, при чому n - степінь числа 2. Тоді можно розбити кожну матрицю на чотири (n/2)\*(n/2) матриці й через них виразити добуток матриць A і B.

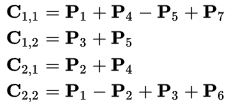


Визначимо нові елементи:



Таким чином, необхідно лише 7 множень на кожному етапі рекурсії.

Елементи матриці С виражаються з Pk за формулами:



Рекурсивний процес продовжується n разів, доти, доки розмір матриць Сi,j не стане достатньо малим, далі використовують звичайний алгоритм множення матриць. Це роблять через те, що алгоритм Штрассена втрачає ефективність порівняно зі звичайним на малих матрицях у силу більшої кількості додавань.

Експериментальним шляхом я визначив, що для мого комп’ютера, вже при n >= 32 алгоритм Штрассена давав виграш порівняно зі звичайним. (Рекурсивне розкладання матриць виконується до досягнення порядку n = 16, після цього використовується звичайне множення; для того, щоб алгоритм Штрассена виконав хоча б один крок рекурсії, необхідно щоб n початкових матриць = 16\*2 = 32. При розкладанні до n = 8 алгоритм Штрассена сповільнював множення).

Програма виводить час, витрачений на виконання алгоритму Штрассена, і після нього, час, витрачений на виконання звичайного алгоритму.

Складність алгоритму: час – O (Nlog2 7 + o(1)), що є приблизно О(N2,8074)

Реалізація (Python 3.8)