Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних інформаційних систем

Алгоритми та складність

Завдання №2

“Алгоритм Штрассена”

Виконав студент 2-го курсу

Групи К-28

Гуща Дмитро Сергійович

2020

**Завдання**

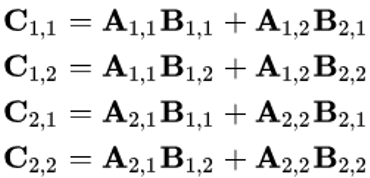
Реалізуйте алгоритм Штрассена для множення матриць. На практиці алгоритм починає застосовуватися для матриць такого розміру, коли з'являється виграш порівняно з класичним способом на основі означення, який використовується для матриць меншого розміру. Спробуйте експериментально визначити цю "точку перетину" для свого комп'ютера.

**Теорія**

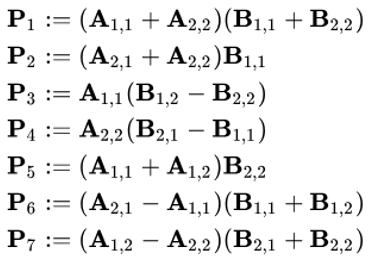
Нехай А, В – дві квадратні матриці над кільцем R. Матриця С знаходиться за формулою Якщо розмір матриць які множаться n не є натуральною степінню двійки, ми доповнюємо початкові матриці додатковими нульовими строками та стовбцями. При цьому ми отримуємо зручні для рекурсивного множення розміри, але втрачаємо в ефективності за рахунок додаткових непотрібних множень. Розділимо матриці А, В і С на рівні за розмірои блочні матриці



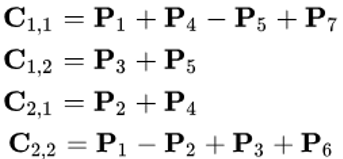
Де



Проте за допомогою цієї процедури нам не вдалось зменшити кількість множень. Як і в звичайному методі, нам потрібно 8 множень. Тепер визначимо нові елементи які потім користують для визначення Ci,j. Таким чином, нам потрібно всього 7 множень на кожному етапі рекурсії.



Елементи матриці С визначаються по формулам



Рекурсивний процес продовжується n разів доти, поки розмір матриці Ci,j не стане достатньо малим, далі використовують звичайний метод множення матриць. Це роблять через те, що алгоритм Штрассена втрачає ефективність в порівнянні зі звичайним на малих матрицях через велике число сумувань. Оптимальний розмір матриці для переходу до звичайного множення залежить від характеристик процесора і мови програмування і на практиці лежить в границях від 32 до 128

**Складність**

Стандартне множення матриць займає приблизно 2N, (де N = 2n) арифметичних операцій(сумування і множення); асимптотична складність О(N3). Число сумувань і множень, необхідних в алгоритмі Штрассена може бути розрахована наступним чином. Нехай f(n) число операцій для матриці 2n × 2n . Тоді по рекурсивним застосуванням алгоритма Штрассена, ми бачимо, що f(n) = 7f(n-1) + L4n , з деякою константою L, яка залежить від кількості доповнення, виконаних в кожному застосуванні алгоритма. Отже f (n) = (7 + o (1))n , тобто асимптотична складність для множення матриць розміру N = 2n  використовуючи алгоритм Штрассена є



Зменшення кількості арифметичних операцій приводить до частинно зменшеної числової стабільності і алгоритм також потребує значно більше пам’яті на відміну від наївного алгоритму. Обидві начальні матриці, розміри яких повинні бути розширені до наступної ступені двійки, в результаті чого зберігається до чотирьох разів більше елементів і сім допоміжних матриць, кожна з яких містить в собі чверть елементів.

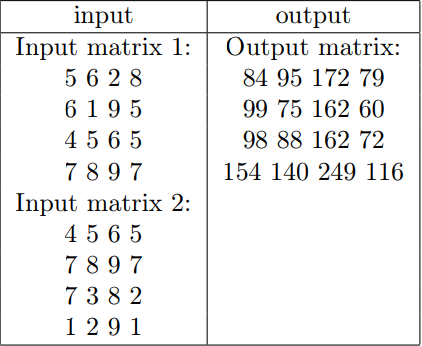
**Мова програмування**

С++

**Модулі програми**

void СoutMatrix(int n, int C[][N])  
//Виводить матрицю.  
void СlassicMultiply(int A[][N], int B[][N], int C[][N])  
//Множення матриць.  
void Add(Substract)Matrix(int n, int X[][N], int Y[][N], int Z[][N])  
//Додавання(Вiднiмання) матриць.  
void Strassen(int n, int A[][N], int B[][N], int C[][N])  
//Сам алгоритм Штрассена.

**Тестові приклади**



**Висновок**

Алгоритм Штрассена пiдходить для швидких обчислень великих матриць. Для матриць, розмiр яких укладається в промiжки, використання алгоритму Штрассена дозволить прискорити процес множення. Звичайно, множити такi матрицi доводиться не часто, але прикладних задач, в яких необхiдно перемножати, наприклад, матрицi 4000 на 4000, для яких корисний даний алгоритм, не мало.

**Лiтература**

Дж. Макконнел. Основы современных алгоритмов.

А. В. Левитин. Алгоритмы: введение в разработку и анализ.

https://en.wikipedia.org/wiki/Strassen-algorithm