Nguyễn Ngọc Trà My – 4.01.103.068

**Bài thực hành**

Bài thực hành 3: Cài đặt giải thuật Strassen cho nhân đa thức

**Input:** Hai đa thức A và B

**Output:** Kết quả của phép nhân đa thức A và B

**Cách xử lý**

* Kiểm tra kích thước của hai đa thức A và B. Nếu kích thước nhỏ hơn một giá trị cơ sở nào đó (thường là 1 hoặc 2), ta thực hiện phép nhân đa thức thông thường và trả về kết quả.
* Nếu kích thước lớn hơn giá trị cơ sở, ta chia đa thức A và B thành các đa thức con. Thường là chia theo cách chia đa thức ban đầu thành bốn đa thức con có kích thước bằng nhau.
* Sử dụng đệ quy tính toán các đa thức con bằng cách áp dụng lại giải thuật Strassen trên các đa thức con này.
* Từ các kết quả đa thức con, ghép chúng lại để tạo thành đa thức kết quả.
* Trả về đa thức kết quả.

**Code**

#include <iostream>

#include <vector>

using namespace std;

// Hàm nhân đa thức theo giải thuật Strassen

vector<int> strassenMultiplication(const vector<int>& A, const vector<int>& B) {

int n = A.size();

// Trường hợp cơ sở: đa thức bậc 0

if (n == 1) {

vector<int> C(1);

C[0] = A[0] \* B[0];

return C;

}

// Chia đa thức A và B thành các đa thức con

int halfSize = n / 2;

vector<int> A0(A.begin(), A.begin() + halfSize);

vector<int> A1(A.begin() + halfSize, A.end());

vector<int> B0(B.begin(), B.begin() + halfSize);

vector<int> B1(B.begin() + halfSize, B.end());

// Tính các đa thức con theo công thức Strassen

vector<int> M0 = strassenMultiplication(A0, B0);

vector<int> M1 = strassenMultiplication(A0, B1);

vector<int> M2 = strassenMultiplication(A1, B0);

vector<int> M3 = strassenMultiplication(A1, B1);

// Tính đa thức kết quả theo công thức Strassen

vector<int> C(n, 0);

for (int i = 0; i < halfSize; i++) {

C[i] += M0[i];

C[i + halfSize] += M1[i] + M2[i];

C[i + n] += M3[i];

}

return C;

}

int main() {

vector<int> A = {1, 2, 3, 4}; // Đa thức A

vector<int> B = {5, 6, 7, 8}; // Đa thức B

vector<int> C = strassenMultiplication(A, B); // Nhân đa thức A và B

cout << "Ket qua: ";

for (int element : C) {

cout << element << " ";

}

cout << endl;

return 0;

}

**Kết quả**

A computer screen with green text

Description automatically generated with low confidence