CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

BÁO CÁO BÀI TẬP NHÓM G-2

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

MỤC LỤC

1

Tổng quan

Thông tin nhóm

Tên nhóm: Nhóm 8

Mã nhóm: G-08

STT	MSSV	Họ tên	Email	SĐT
1	1612364	Nguyễn Hoàng Lưu	1612364@student.hcmus.edu.vn	0978.516.054
2	1612380	Phạm Hoàng Minh	1612380@student.hcmus.edu.vn	0162.909.3399
3	1612382	Huỳnh Nguyễn Nhật Minh	1612382@student.hcmus.edu.vn	0122.385.0377
4	1612392	Nguyễn Thị Hồng Mơ	1612392@student.hcmus.edu.vn	0165.234.9848
5	1612471	Hoàng Thị Hoài Nhi	1612471@student.hcmus.edu.vn	0979.127.705

Thông tin bài tập

Tên bài tập	G-2
Nội dung	1. Thiết kế các cấu trúc dữ liệu cần thiết cho bài toán theo yêu cầu
	2. Khai báo cấu trúc dữ liệu này và commit lên Bitbucket
	+ Chỉ khai báo cấu trúc dữ liệu, không cài đặt hàm hay method.
	+ Nên có hình ảnh minh họa.
	+ Các khai báo nên có comment rõ ràng
	3. Viết báo cáo & nộp bài.

Tổ chức nhóm

Vai trò	Thành viên	Nhiệm vụ
Project Manager	Huỳnh Nguyễn Nhật Minh	Phân chia công việc cho các thành viên khác. Viết báo cáo và nộp bài
Developer	Phạm Hoàng Minh	Mỗi thành viên sẽ tự xây dựng một cấu trúc dữ liệu phù hợp với yêu cầu đề. Khi họp nhóm sẽ cùng thống nhất và chọn ra phương án phù hợp, đạt hiệu quả cao nhất.
Developer	Nguyễn Thị Hồng Mơ	
Developer	Hoàng Thị Hoài Nhi	
Developer	Nguyễn Hoàng Lưu	

2

Tiến độ thực hiện

Thời gian	Công việc thực hiện
19/10/2017	Nhận bài tập từ giáo viên.
20/10/2017	Họp nhóm online. Project Manager phân chia nhiệm vụ cho các thành viên. Cả nhóm phân tích, tìm hiểu đề bài.
21/10/2017 – 25/10/2017	Các thành viên tự xây dựng cấu trúc dữ liệu theo kết quả phân tích được.
26/10/2017	Họp nhóm trực tiếp (gặp mặt).
27/10/2017	Họp nhóm online, sửa chữa lại cấu trúc dựa trên thông tin, gợi ý thầy cung cấp trên lớp.
29/10/2017	Hoàn thành báo cáo, commit source code lên Bitbucket, lưu trữ tài liệu lên drive.

Biên bản họp nhóm lần 1:

https://drive.google.com/open?id=0B0gweI2LgCdhMzM5eUpmVXZqakU

Sau khi họp nhóm lần 1, nhóm G-08 đã thống nhất được cấu trúc dữ liệu sẽ dùng. Tuy nhiên, sau buổi học ngày 26/10/2017, nhóm đã tiếp thu những góp ý của thầy và có những thay đổi cho phù hợp hơn.

Những ghi chú được nhóm em ghi trực tiếp trong báo cáo này, không ghi trong project nộp kèm theo.

3

Báo cáo chi tiết

1. Cấu trúc dữ liệu đã xây dựng và lý giải về lý do chọn:

a) Xây dựng một class DonThuc để lưu một đơn thức, trong đó:

+ Sử dụng kiểu float để lưu hệ số cho đơn thức ➔ với những bài toán cộng, trừ, nhân chia đơn thức thì kiểu float là đủ để lưu giá trị (miền giá trị kiểu float từ khoảng ~10^{-38} đến ~10^{38}).

+ Sử dụng một danh sách liên kết để lưu các kí tự biến số ➔ có thể lưu được bao nhiêu số tùy thích (mở rộng dễ dàng, đặc biệt là trong trường hợp cần nhân nhiều đơn thức – đa thức lại với nhau). Không lãng phí bộ nhớ (khi nào cần sử dụng mới tạo thêm)

+ Sử dụng một danh sách liên kết để lưu giá trị của các số mũ ỨNG VỚI giá trị của các biến (lưu lần lượt theo thứ tự).

b) Xây dựng một class DaThuc để lưu một đa thức (bao gồm nhiều đơn thức), trong đó:

+ Sử dụng danh sách liên kết để lưu đa thức (mỗi Node của danh sách liên kết sẽ lưu một đơn thức)

➔ Sử dụng danh sách liên kết sẽ giúp việc rút gọn đa thức (xóa phần tử) đơn giản hơn (nếu so với mảng, vì mảng cần tốn chi phí dịch chuyển phần tử).

➔ Ngoài ra, việc sắp xếp đa thức cũng tốn khá ít chi phí nếu dùng danh sách liên kết đơn (có thể dùng Merge Sort để sắp, tốn O(nlogn))

Cấu trúc dữ liệu như sau:

```
// Danh sách liên kết dùng để lưu biến hoặc số mũ.

template <class T>
struct Node_Bien_Mu
{
        T data;             // khi dùng để lưu biến, data sẽ mang kiểu char, khi dùng lưu
                            // số mũ sẽ mang kiểu float (để lưu được số mũ phân số)
        Node_Bien_Mu* next;
};

template <class T>
class DonThuc
{
        float  _HeSo;
        Node_Bien_Mu*  _Bien;  // sử dụng danh sách liên kết để lưu được nhiều kí
                               // tự hơn
        Node_Bien_Mu*  _SoMu;
        DonThuc* _Next;         // trỏ đến đơn thức tiếp theo
public:
};


template< class T>
class DonThuc;

template <class T>
class DaThuc
{
        DonThuc* _Donthuc;      // lưu giá trị của node đầu tiên của danh sách liên kết
                                // gồm các đơn thức
public:
};
```