LỜI NÓI ĐẦU

Cấu trúc dữ liệu và giải thuật là học phần rất quan trọng đối với người lập trình. Môn học này được xem như là nền tảng của lập trình máy tính. Nó là cơ sở vững chắc để giải quyết một số bài toán , đồng thời cung cấp cho chúng ta hiểu biết về các giải thuật tác động lên dữ liệu như thế nào, cũng như tổ chức dữ liệu để giải quyết các bài toán sao cho hiệu quả nhất, tối ưu hóa nhất. Và cũng là môn học tạo ra môi trường khởi đầu để các em sáng tạo.

Chúng em nghiên cứu và thực hiện đồ án này như một cách để cũng cố và mở rộng kiến thức về môn cấu trúc dữ liệu. Thông qua quá trình thực hiện đồ án, chúng em đã nắm bắt được những kỹ thuật quan trọng của việc xây dựng cấu trúc dữ liệu và cách xây dựng thuộc toán sao cho tối ưu nhất.

Bài Toán “ Sắp xếp danh sách liên kết đơn bằng phương pháp merge sort” mà chúng em nghiên cứu và trình bày trong báo cáo sau đây là một trong các ví dụ của môn cấu trúc dữ liệu và giải thuật.

Đề bài

“Sắp xếp danh sách liên kết bằng phương pháp merge sort”

Khởi tạo một danh sách liên kết đơn ngẫu nhiên gồm n phần tử với n nhập từ bàn phím. Hãy dùng phương pháp sắp xếp merge sort ( sắp xếp bằng con trỏ ID) để sắp xếp sao cho danh sách liên kết đơn giảm dần.

Lý do chọn đề tài

Các kiểu cấu trúc dữ liệu như danh sách liên kết đơn, ngăn xếp, hàng đợi,cây nhị phân tìm kiếm…cùng với các giải thuật chiếm một vị thế rất quan trọng tronhg khoa học máy tính. Ngày nay , với sự phá triển của công nghệ thông tin, các thuật toán mới ra đời giúp cho con người giải quyết những bài toán phức tạp. nhưng vai trò của kiểu cấu trúc dữ liệu link list không hề giảm bớt, nó chính là kiểu dữ liệu cơ sở để áp dụng các thuật toán phức tạp khác. Nhờ có danh sách liên kết mà một số bài toán phúc tạp được giải quyết một cách dễ dàng hơn.Trong kiểu link list thì các thuật toán sắp xếp luôn thông dụng, nó có thể giải quyết một số bài toán liên quan đến thục tế như quản lý nhân viên, quản lý một chuỗi cung ứng dịch vụ nào đó mà đòi hỏi phải có sự sắp xếp theo một tiêu chí nào đó để thuận tiện trong việc quản lý.Một số phương pháp sắp xếp danh sách liên kêt có thể kể ra như selection sort, interchange sort,insertion sort, build sort….Nhưng điều quan trọng với một lượng dữ liệu cực lớn thì các phương pháp sắp xếp trên có còn hiệu quả, có còn đảm bảo tiết kiệm chi phí hay không… Merge sort là cách giải quyết có thể nói là tối ưu trong việc dữ liệu cực lớn nó có thể sắp xếp nhanh rất nhiều lần so với các phương pháp kể trên. Không những thế bài toán “ sắp xếp danh sách liên kết bằng phương pháp merge sort” có thể giúp chúng em tìm hiểu sâu hơn về môn cấu trúc dữ liệu và giải thuật. Chính vì nhứng lý do trên nhóm em chọn đề tài cho đồ án là “ sắp xếp danh sách liên kết đơn bằng phương pháp merge sort”

Mục tiêu và nhiệm vụ

* Nghiên cứu và viết code để thấy rõ tác vai trò của merge sort trong danh sách liên kết đơn.
* Hiểu rõ bản chất của thuật toán, quy trình hoặc động của merge trong danh sách liên kết đơn
* Hoàn thành tất cả các quy trình của thuật toán, có thể nghiên cứu và phát triển thêm những phần liên quan đến thuật toán.

Đối tượng nghiên cứu

* Lý thuyết về cấu trúc dữ liệu trừu tượng link list
* Cơ chế hoạt động của danh sách liên kết và một số thao tác cơ bảng trên danh sách liên kết đơn
* Ngôn ngữ lập trình Visual studio C/C++ dùng để phục vụ cho việc hoàn thành bài toán .

Phần định nghĩa:

Lý thuyết về Danh sách liên kết đơn

Cũng giống như mảng danh sách liên kết đơn cũng bao gồm các phần tử nhưng có liên kết vơi nhau bởi các địa chỉ gọi tắt là con trỏ.

Mỗi phần tử của danh sách liên kết đơn là một kiểu cấu trúc chứa 2 thông tin

* Thành phần dữ liệu: lưu trữ thông tin cơ bản của phần của phần tử có kiểu dữ liệu là int,char,…. Tùy vào mục đich của bài toán.
* Thành phần liên kết: lưu trữ địa chỉ của phần tử kế tiếp nó trong danh sách liên kết , hoặc lưu trữ giá trị NULL nếu như phần tử đó là phần tử cuối cùng của danh sách liên kết đơn.
* Ta có định nghĩa tổng quát

Typedef struct Node

{

data Info; //data là kiểu dữ liệu từ ý mà người tạo muốn có để xử lý bài toán

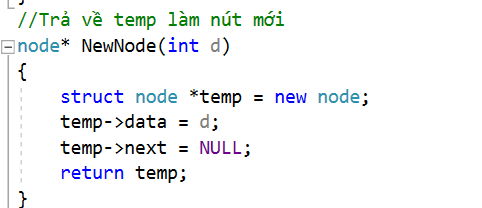
Struct Node\* pnext;  
 };

* Một phần tử trong danh sách đơn là một biến động được yêu cầu cấp phát khi cần và danh sách đơn chính là sự liên kết các biến động này lại với nhau
* Nếu biết địa chỉ của phần tử đầu tiên trong danh sách liên kết đơn ta có thể truy xuất đến phần tứ thứ 2 dựa vào thông tin của pnext.
* Về danh sách liên kết đơn ta có 2 biến địa Node\* phead và Node \* ptail để quản lý phần tử đầu tiên và phần tử cuối cùng của danh sách liên kết đơn

PHẦN ĐỒ ÁN

Phân tích thiết kế thuật toán

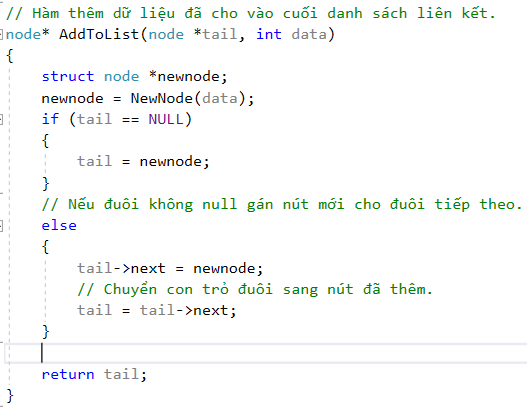
1. Hàm tạo một node mới.



* Hàm sẽ trả về một Node mới với thông tin dữ liệu là một số nguyên. Ban đầu ta xem như danh sách chưa có một phần tử nào muốn có phần tử để tiến hành sắp xếp ta phải khởi tạo 1 Node mới bằng cách khai báo 1 struct node\*temp. chúng ta có thể dùng con trỏ để trỏ đến data của nó gán nó bằng giá trị d mà mình tạo từ hàm ngẫu nhiên ở trên.
* Chúng ta xem như phần tử vừa tạo là 1 phần tử đơn nên temp 🡪 next =NULL
* Sau khi làm xong ta trả về giá trị của biến temp.

1. Hàm thêm Node vào danh sách liên kết đơn

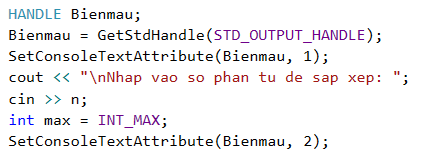
* Ở đây chúng em thực hiện hàm thêm Node vào cuối danh sách.



* Ở Hàm thêm Node vào cuối danh sách liên kết đơn đầu tiên chúng ta phải gọi lại hàm khởi tạo Node ở trên để ta có 1 Node hoàn chỉnh.
* Khi đã có Node ta tiến hành thêm Node vào cuối danh sách bằng cách kiểm tra xem danh sách đó có rỗng hay không nếu rỗng thì xem như khi thêm Node vừa mới khởi tạo vào mặc nhiên nó là phần tử duy nhất trong danh sách và nó nằm ở vị trí cuối cùng trong danh sách (tail).
* Ngược lại nếu trong danh sách đã có ít nhất 1 phần tử thì ta tiến hành gán con trỏ next của phần tử cuối cùng trong danh sách đến biến con trỏ next của phần tử ta muốn thêm vào.
* Tiếp theo ta gán lại biến con trỏ tail vào Node đã thêm. Và cuối cùng là trả về giá trị của Node mới thêm vào tức là tail.
* Kết quả là Node vừa mới thêm vào sẽ có giá trị là data nằm cuối cùng danh sách và biên tail 🡪next sẽ là NULL.

1. Hàm tạo màu bổ Trợ ( không bắt buộc phải có)

* Để làm được code màu trước hết ta phải khai báo thư viện windows.h thư viện này cho phép ta có thể sử dụng một số code màu.



* Ta khai báo một biến màu Bienmau có kiểu dữ liệu HANDLE
* Gọi hàm GetStdHandle(STD\_OUTPUT\_HANDLE) này cho phép ta sử dụng 255 màu có sẵn trong thư viện Windows
* Tiếp đến ta gọi hàm SetConsoleTextAttribute để tô màu cho những dòng chữ xuất ra ở đây chọn màu 1 để in dòng chữ “Nhap vao so phan tu de sap xep: ”

