Chương 2

BÀI TẬP LÝ THUYẾT

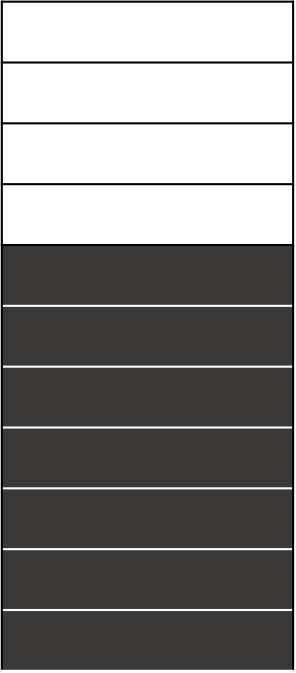
**Câu 1**: Trong khoa học máy tính, **danh sách đặc** được hiểu như thế nào? Cho ví dụ

**Trả lời:** Danh sách đặc là một danh sách mà các phần tử trong danh sách có **cùng kiểu dữ liệu**, và được **cấp phát liên tục** trong bộ nhớ.

**Ví dụ:**

‘MAX’ là độ dài tối đa của danh sách đặc

MAX - 1

****

Hiện đang lưu trữ n phần tử

a[n-1] là biến chứa giá trị/dữ liệu của danh sách

tại vùng có chỉ số n-1

a[1] là biến chứa giá trị/dữ liệu của danh sách

tại vùng có chỉ số 1

0, 1, 2, 3…: là chỉ số từng phần tử trong danh sách

a[0] là biến chứa giá trị/dữ liệu của danh sách

tại vùng có chỉ số 0

n - 1

a[n - 1]

…

…

…

a[2]

a[1]

a[0]

0

1

2

3

…

…

…

**Câu 2 :** Trong khoa học máy tính, **danh sách liên kết** được hiểu như thế nào? Có mấy loại? Cho ví dụ.

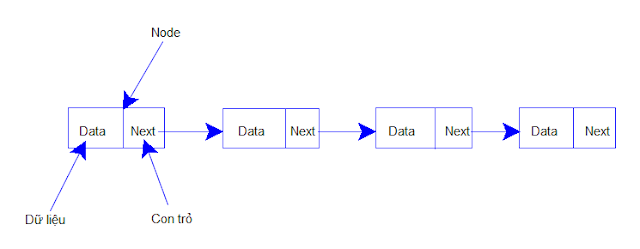
**Trả lời:** Danh sách liên kết đơn là một ***danh sách*** mà các phần tử được cấp phát *rời rạc* nhau, và cố định trong bộ nhớ. Mỗi Phần tử trong danh sách gồm có **2 thành phần:**

* Phần 1: vùng thông tin chưa giá trị cần quả lý
* Phần 2: vùng liên kết, chứa địa chỉ bộ nhớ của phần tử kế tiếp

Có 2 loại:

* Danh sách liên kết đơn
* Danh sách liên kết kép

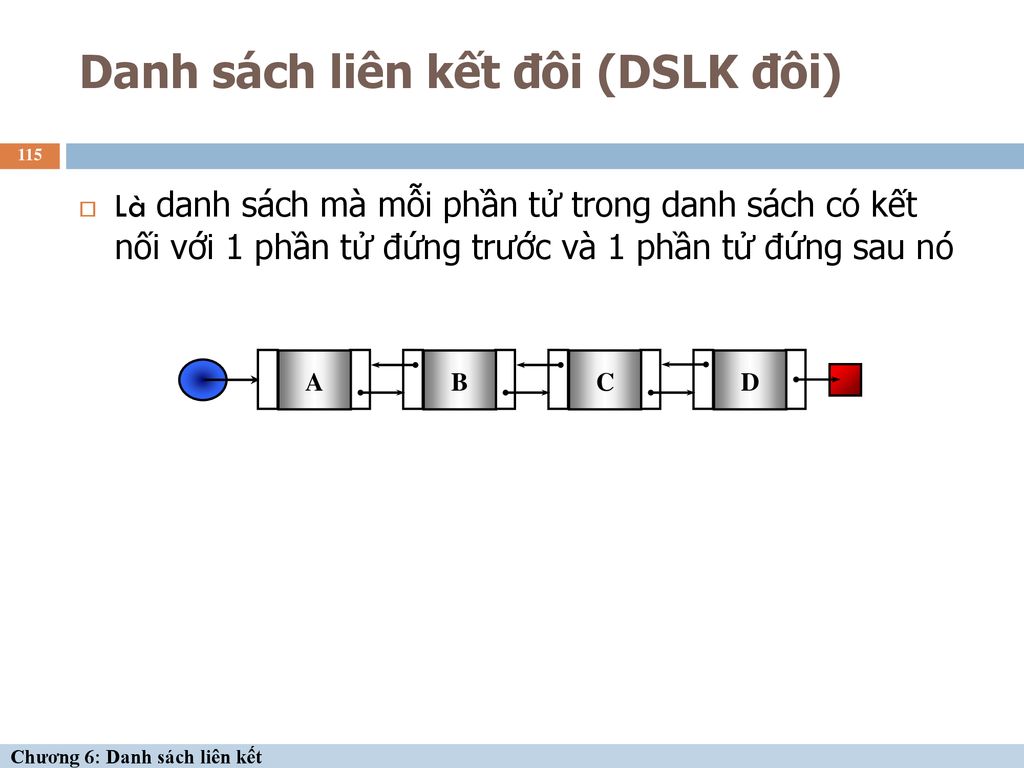
Ví dụ : Danh sách liên kết đơn



Ví dụ : Danh sách liên kết kép

**Next**

**NULL**



**Previous**

**NULL**

**Câu 3 :** Tại sao nói **STACK** và **QUEUE** là danh sách hạn chế? Cho ví dụ?

Vì STACK và QUEUE là một danh sách các phần tử được quản lí theo thứ tự như sau:

Ngăn xếp (Stack) là một danh sách các phần tử được quản lý theo thứ tự như sau: Phần tử được *thêm vào* ngăn xếp *sau*, sẽ *được lấy* ra (xóa) khỏi ngăn xếp *trước.*

Hàng đợi (Queue) là danh sách chứa các phần tử được quản lý theo thứ tự sau: Phần tử được *thêm vào* trước, sẽ *được lấy* ra (xóa) *trước*

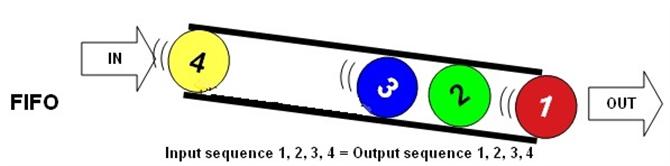
* Chúng bị hạn chế vì phải tuân theo những nguyên tắc : Thêm vào và lấy ra có quy tắc.

VD: (STACK) có thể hiểu là một chồng bát, bạn chồng các chiếc bát lên cao thì chiếc bạn chồng vào sau cùng sẽ là chiếc bạn lấy ra đầu tiên và ngược lại, chiếc bát đầu tiên, ở dưới cùng sẽ là chiếc bạn lấy ra sau cùng

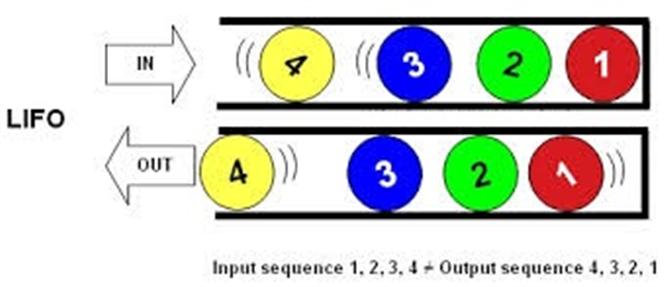
(QUEUE) có thể liên tưởng đến lúc bạn xếp hàng mua vé xem phim, ai đến xếp hàng trước thì sẽ được bán vé trước và tương tự, ai xếp hàng sau thì sẽ được phục vụ sau.

**Câu 4 :** Thế nào là **LIFO**, **FIFO** ? Cho ví dụ

**FIFO (First in First out –**Vào trước Ra trước): Với phương pháp này, các lô hàng đầu tiên của hàng hoá nhập vào nhà kho sẽ là hàng hoá đầu tiên được xuất ra khỏi kho – từ đó được gửi vào các cửa hàng hoặc gửi trực tiếp đến khách hàng.



**LIFO (Last in First Out – Vào sau Ra trước):**Ở phương pháp này, các hàng hoá được nhập vào kho gần nhất sẽ được xuất ra đầu tiên. Hàng hoá mới được sử dụng trước, dùng ưu tiên hơn hàng hoá cũ.



**Câu 6 :** Theo bạn, danh sách danh sách liên kết có thể ứng dụng xử lý các vấn đề gì trong máy tính?

Trả lời: Khắc phục tình trạng phân mảnh bộ nhớ.

Không cần cấp phát trước số lượng như bên mảng, dữ liệu luôn được ở trạng thái động (Khi nào cần thêm vào thì cứ thêm vào như bình thường, không cần phải cấp lại bộ nhớ như bên mảng).

Các thao tác thêm / xóa không cần phải lùi lại hay tiến lên các phần tử từ vị trí đó trở xuống như bên mảng mà ta chỉ cần bẻ lại các mối liên kết giữa 2 Node là được => Chạy nhanh hơn bên mảng trong trường hợp này.

**Câu 7** : Thế nào là cấu trúc dữ liệu động? Cho ví dụ.

**What is Dynamic Data Structure?**  
In Dynamic data structure the size of the structure in not fixed and can be modified during the operations performed on it. Dynamic data structures are designed to facilitate change of data structures in the run time.( Trong cấu trúc dữ liệu động, kích thước của cấu trúc không cố định và có thể được sửa đổi trong các hoạt động được thực hiện trên nó. Cấu trúc dữ liệu động được thiết kế để tạo điều kiện thay đổi cấu trúc dữ liệu trong thời gian chạy.)

**VD:** 1 .Trong thực tế, một số đối tượng có thể được định nghĩa đệ qui, ví dụ để mô tả đối tượng "con người" cần thể hiện các thông tin tối thiểu như :

     Họ tên

     Số CMND

     Thông tin về cha, mẹ

Ðể biễu diễn một đối tượng có nhiều thành phần thông tin như trên có thể sử dụng kiểu bản ghi. Tuy nhiên, cần lưu ý cha, mẹ của một người cũng là các đối tượng kiểu NGƯỜI, do vậy về nguyên tắc cần phải có định nghĩa như sau:

typedef  struct NGUOI{

char  Hoten[30];

int   So\_CMND ;

NGUOI Cha,Me;

};

Nhưng với khai báo trên, các ngôn ngữ lập trình gặp khó khăn trong việc cài đặt không vượt qua được như xác định kích thước của đối tượng kiểu NGUOI.

2.       Một số đối tượng dữ liệu trong chu kỳ sống của nó có thể thay đổi về cấu trúc, độ lớn, như danh sách các học viên trong một lớp học có thể tăng thêm, giảm đi ... Khi đó nếu cố tình dùng những cấu trúc dữ liệu tĩnh đã biết như mảng để biểu diễn những đối tượng đó lập trình viên phải sử dụng những thao tác phức tạp, kém tự nhiên khiến chương trình trở nên khó đọc, do đó khó bảo trì và nhất là khó có thể sử dụng bộ nhớ một cách có hiệu quả.

3.      Một lý do nữa làm cho các kiểu dữ liệu tĩnh không thể đáp ứng được nhu cầu của thực tế là tổng kích thước vùng nhớ dành cho tất cả các biến tĩnh chỉ là 64Kb (1 Segment bộ nhớ). Khi có nhu cầu dùng nhiều bộ nhớ hơn ta phải sử dụng các cấu trúc dữ liệu động.

4.       Cuối cùng, do bản chất của các dữ liệu tĩnh, chúng sẽ chiếm vùng nhớ đã dành cho chúng suốt quá trình hoạt động của chương trình. Tuy nhiên, trong thực tế, có thể xảy ra trường hợp một dữ liệu nào đó chỉ tồn tại nhất thời hay không thường xuyên trong quá trình hoạt động của chương trình. Vì vậy việc dùng các CTDL tĩnh sẽ không cho phép sử dụng hiệu quả bộ nhớ.

Do vậy, nhằm đáp ứng nhu cầu thể hiện sát thực bản chất của dữ liệu cũng như ?ây dựng các thao tác hiệu quả trên dữ liệu, cần phải tìm cách tổ chức kết hợp dữ liệu với những hình thức mới linh động hơn, có thể thay đổi kích thước, cấu trúc trong suốt thời gian sống. Các hình thức tổ chức dữ liệu như vậy được gọi là cấu trúc dữ liệu động. Chương này sẽ giới thiệu về các cấu trúc dữ liệu động và tập trung khảo sát cấu trúc đơn giản nhất thuộc loại này là danh sách liên kết.