CẤU TRÚC MẢNG VÀ DANH SÁCH TUYẾN TÍNH

Phần 2: Cấu trúc danh sách

Nội dung chính

- Giới thiệu cấu trúc danh sách
- Cấu trúc vào sau ra trước (LIFO) (Stack-Ngăn xếp)
- Cấu trúc vào trước ra trước (FIFO) (Queue-Hàng đợi)
- Một số ứng dụng của ngăn xếp và hàng đợi

Giới thiệu cấu trúc danh sách

- Danh sách tuyến tính: CTDL gồm một hay nhiều phần tử cùng kiểu dữ liệu và tồn tại một trật tự tuyến tính giữa các phần tử.
- Kí hiệu: L = <x1,x2,...,xn>
  - $n \geq 1$ và x1,x2,...,xn là các phần tử của danh sách,
  - x1 được gọi là phần tử đầu tiên (đầu) của danh sách
  - xn được gọi là phần tử cuối cùng (đuôi) của danh sách
- Trật tự tuyến tính: trật tự trước-sau giữa các phần tử, tức là với mọi cặp phần tử <xi,xj> ($1 \leq i, j \leq n$ và $i \neq j$) trong tập các phần tử này luôn có duy nhất một trật tự trước sau.
- Quy ước: trường hợp đặc biệt khi danh sách không có phần tử nào, gọi là danh sách rỗng, kí hiệu $\varnothing$ ($L = \varnothing$).

Giới thiệu cấu trúc danh sách

- Kích thước hay độ dài danh sách:
  - Là số phần tử của danh sách
  - Kích thước của danh sách rỗng bằng 0
  - Không cố định mà biến đổi trong quá trình xử lý, thao tác và nó là đại lượng mà ta thường không biết trước được.
- Kiểu dữ liệu của các phần tử:
  - Có một kiểu dữ liệu duy nhất cho các phần tử của danh sách
  - Kiểu dữ liệu cho các phần tử luôn luôn cố định.
- Trật tự tuyến tính trong danh sách:
  - Một danh sách luôn có hai phía, một phía được quy ước là đầu, còn phía kia là đuôi của danh sách.
  - Trật tự trước-sau là trật tự từ đầu đến cuối.

Các thao tác cơ bản trên danh sách

- Khởi tạo danh sách
  - Định ra cấu trúc danh sách, xác định các đặc trưng của danh sách.
  - Sau khởi tạo, ta có một danh sách rỗng (chưa có nội dung, chỉ có phần khung).
  - Trong các ngôn ngữ lập trình, thao tác khởi tạo thường là việc khai báo cấu trúc lưu trữ thích hợp để biểu diễn cho cấu trúc danh sách yêu cầu.
- Bổ sung một phần tử mới vào danh sách
  - Trước tiên cần xác định vị trí trong danh sách mà phần tử mới sẽ được đưa vào.
  - Vị trí bổ sung thường là đầu hoặc cuối danh sách. Tuy nhiên, cũng có thể chèn phần tử mới vào giữa danh sách.
  - Mỗi thao tác bổ sung làm tăng kích thước danh sách lên 1.
  - Chú ý, điều kiện để thực hiện là danh sách chưa "đầy" hay chưa bão hoà.

Các thao tác cơ bản trên danh sách

- Xóa một phần tử khỏi danh sách
  - Thực hiện ngược lại với thao tác bổ sung.
  - Kích thước của danh sách sẽ giảm đi 1.
  - Chú ý, điều kiện thực hiện là danh sách phải không rỗng
- Tìm kiếm một phần tử trong danh sách
  - Tìm kiếm sự xuất hiện hay không của một hay một số phần tử trong danh sách theo một điều kiện tìm kiếm
  - Nếu có, trả về vị trí của phần tử cần tìm trong danh sách.
- Sắp xếp danh sách: sắp xếp các phần tử của danh sách theo một trật tự nhất định
- Nối hai hay nhiều danh sách con để thành một danh sách
- Tách danh sách thành hai hay nhiều danh sách con

Các cấu trúc danh sách thông dụng

- Cấu trúc vào sau ra trước (Last In, First Out - LIFO) hay cấu trúc ngăn xếp – Stack
- Cấu trúc vào trước ra trước (First In, First Out - FIFO) hay cấu trúc hàng đợi – Queue
- Cấu trúc hàng đợi có ưu tiên - Priority queue
  - Mỗi phần tử có thêm một thuộc tính: độ ưu tiên.
  - Với các phần tử có độ ưu tiên như nhau thì danh sách hoạt động giống như nguyên tắc hàng đợi.
  - Khi hai phần tử bất kỳ có độ ưu tiên khác nhau thì danh sách luôn ưu tiên phần tử có độ ưu tiên cao hơn, nên dù phần tử có độ ưu tiên cao hơn có được bổ sung vào sau phần tử có độ ưu tiên thấp hơn thì nó vẫn được loại bỏ ra trước.
  - Cấu trúc này kết hợp cả hai nguyên tắc hoạt động của ngăn xếp và hàng đợi không ưu tiên.

Phương pháp cài đặt cấu trúc danh sách

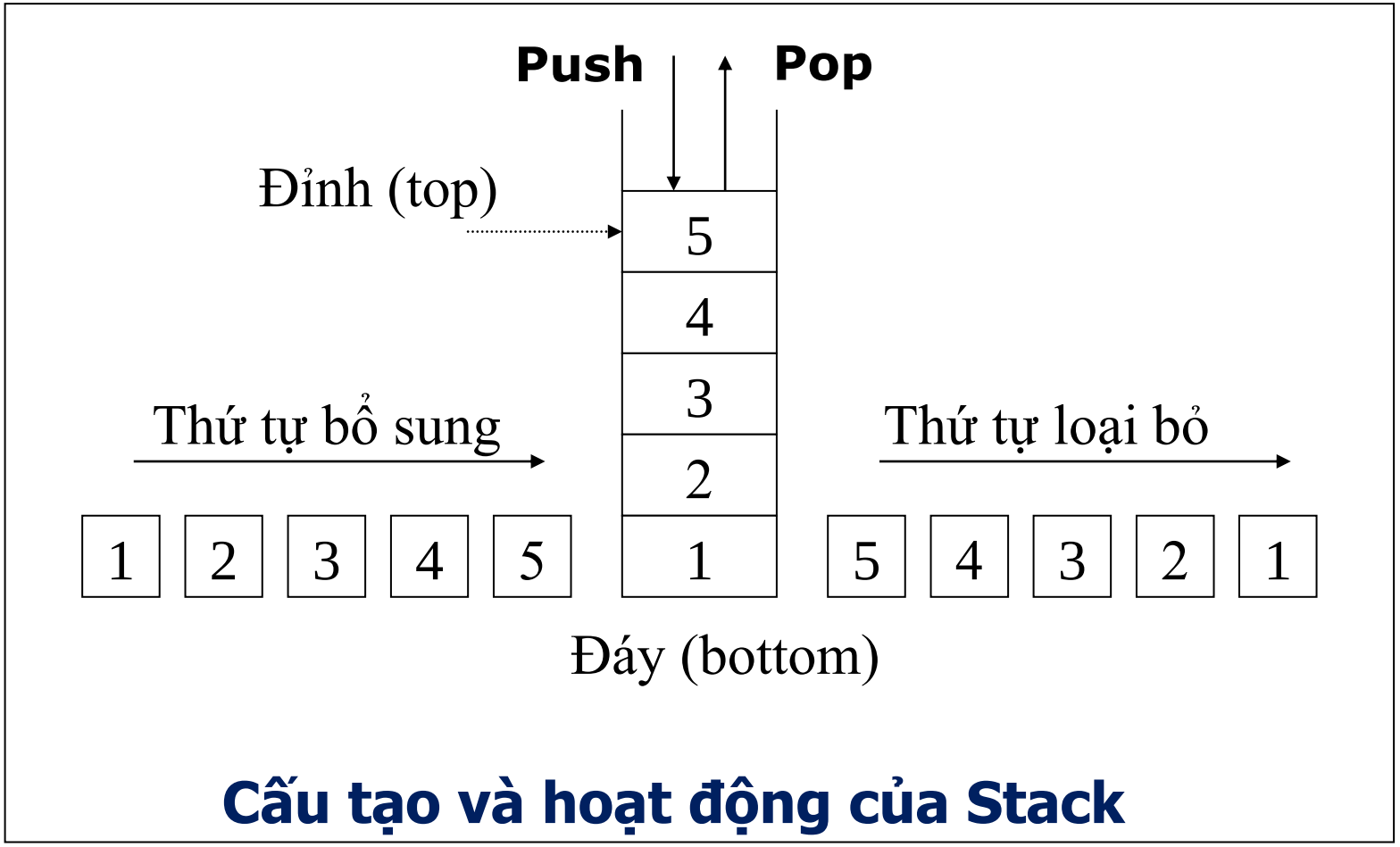
- Nguyên tắc cài đặt:
  - Tính đầy đủ: CTLT phải có các đặc trưng thích hợp để biểu diễn đầy đủ các đặc tính và khả năng của CTDL, thể hiện ở hai khía cạnh
    - Khả năng lưu trữ:
      - là kích thước của CTLT được cài đặt.
      - được quyết định bởi miền giá trị của cấu trúc dữ liệu mà nó cài đặt.
    - Khả năng xử lý: tập các thao tác được cài đặt trên CTLT đã chọn.
  - Tính tối ưu: hiệu quả cao nhất về lưu trữ và xử lý.

Phương pháp cài đặt cấu trúc danh sách

- Cài đặt bằng cấu trúc lưu trữ tuần tự (mảng một chiều):
  - là cách cài đặt đơn giản và nhanh nhất
  - có một số hạn chế do sự khác nhau cơ bản giữa cấu trúc danh sách và cấu trúc mảng, giữa một bên là cấu trúc động, một bên là cấu trúc tĩnh.
- Cài đặt bằng cấu trúc lưu trữ móc nối:
  - là một cấu trúc lưu trữ động với kích thước và tổ chức lưu trữ có thể biến đổi linh hoạt theo yêu cầu của cấu trúc dữ liệu,
  - thích hợp để tổ chức và lưu trữ các cấu trúc dữ liệu động.

Cấu trúc LIFO (Stack)

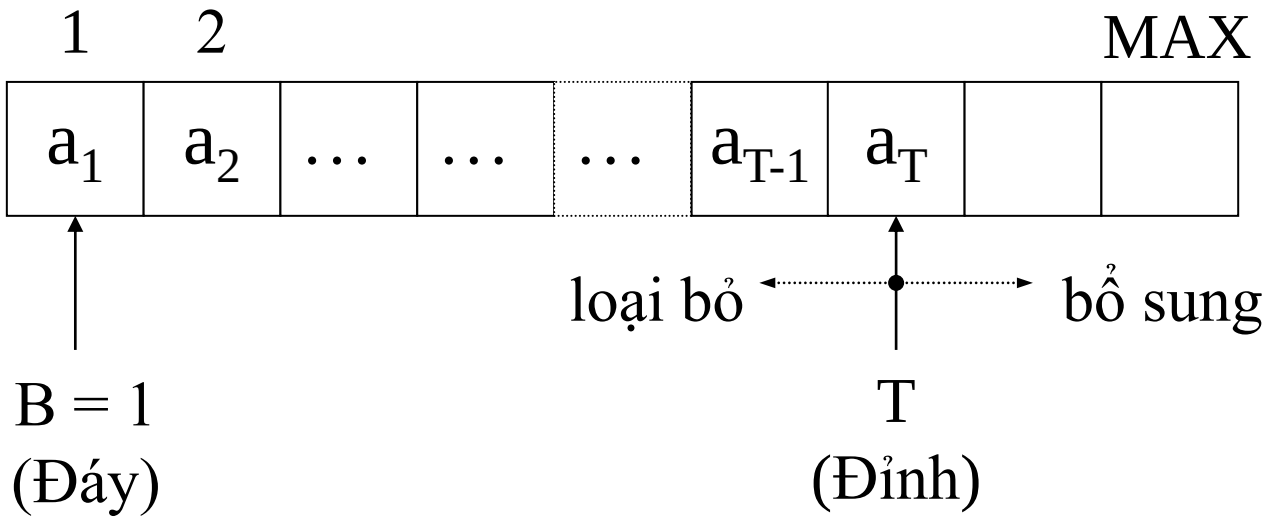
- Mô tả: cấu trúc vào sau ra trước (Last In, First Out - LIFO)



Cấu tạo và hoạt động của Stack

Cài đặt Stack bằng CTLT tuần tự (bằng mảng một chiều)

- Nguyên tắc cài đặt



Biểu diễn ngăn xếp bằng CTLT tuần tự

Cài đặt Stack bằng CTLT tuần tự (bằng mảng một chiều)

```
typedef  struct  {
   Type  info [MAX];
   unsigned int n;          //Số phần tử của Stack
} Stack;
```

- CTLT tuần tự được thể hiện qua mảng info,
- Type là kiểu dữ liệu của danh sách
- n là số phần tử của ngăn xếp.
- Đỉnh của ngăn xếp sẽ ở vị trí n-1
- Trạng thái hiện thời của ngăn xếp:
  - Rỗng (empty): n=0
  - Đầy (full): giá trị MAX là kích thước tối đa của ngăn xếp, ta quy ước ngăn xếp đầy khi n=MAX.
  - Bình thường (normal): trạng thái không rỗng, không đầy. Khi ngăn xếp ở trạng thái này, nó có thể thực hiện các thao tác bổ sung và loại bỏ.

Cài đặt Stack bằng CTLT tuần tự (bằng mảng một chiều)

- Khởi tạo

```
void Initialize (Stack & S){
    S.n =0;
}
```

- Kiểm tra ngăn xếp rỗng

```
bool IsEmpty (Stack S){
    return (S.n == 0);
}
```

- Kiểm tra ngăn xếp đầy

```
bool IsFull (Stack S){
    return (S.n == MAX);
}
```

Cài đặt Stack bằng CTLT tuần tự (bằng mảng một chiều)

- Bổ sung phần tử K vào đỉnh ngăn xếp

```
void Push (Type K, Stack & S){
    if (IsFull(S)) return;
    S.info[S.n] = K;
    S.n++;}
```

- Lấy ra phần tử ở đỉnh ngăn xếp

```
Type Pop (Stack & S){
    if (IsEmpty(S)) return NULL;
    S.n--;
    return S.info[S.n];}
```

- Trả phần tử ở đỉnh ngăn xếp, không lấy ra phần tử đó

```
Type Top (Stack  S){
    if (IsEmpty(S)) return NULL;
    return S.info[S.n - 1];
}
```

Bài tập

1. Cài đặt cấu trúc FIFO (Queue) bằng cấu trúc lưu trữ tuần tự
2. Cài đặt danh sách tổng quát bằng CTLT tuần tự (với danh sách tổng quát, việc bổ sung và lấy ra một phần tử có thể thực hiện ở một vị trí bất kỳ trong danh sách)
3. Viết chương trình chuyển đổi 1 số từ cơ số 10 sang cơ số 2 sử dụng cấu trúc Stack.