**5.1. NGĂN XẾP (STACK)**

Ngăn xếp là một kiểu danh sách được trang bị hai phép toán bổ sung một phần tử vào cuối danh sách và loại bỏ một phần tử cũng ở cuối danh sách.

Có thể hình dung ngăn xếp như hình ảnh một chồng đĩa, đĩa nào được đặt vào chồng sau cùng sẽ nằm trên tất cả các đĩa khác và sẽ được lấy ra đầu tiên. Vì nguyên tắc"vào sau ra trước" đó, Stack còn có tên gọi là danh sách kiểu LIFO (Last In First Out) và vị trí cuối danh sách được gọi là đỉnh (Top) của Stack.

**5.1.1. Mô tả Stack bằng mảng**

Khi mô tả Stack bằng mảng:

Việc bổ sung một phần tử vào Stack tương đương với việc thêm một phần tử vào cuối mảng. Việc loại bỏ một phần tử khỏi Stack tương đương với việc loại bỏ một phần tử ở cuối mảng. Stack bị tràn khi bổ sung vào mảng đã đầy

Stack là rỗng khi số phần tử thực sự đang chứa trong mảng = 0.

program StackByArray;

const

max = 10000;

var

Stack: array[1..max] of Integer;

Last: Integer;

procedure StackInit;

begin

Last := 0;

end;

procedure Push(V: Integer);

begin

if Last = max then WriteLn('Stack is full')

else

begin

Inc(Last); Stack[Last] := V;

end;

end;

function Pop: Integer;

begin

if Last = 0 then WriteLn('Stack is empty')

else

begin

Pop := Stack[Last]; Dec(Last);

end;

end;

begin

StackInit;

<Test>;

end;

Khi cài đặt bằng mảng, tuy các thao tác đối với Stack viết hết sức đơn giản nhưng ở đây ta vẫn chia thành các chương trình con, mỗi chương trình con mô tả một thao tác, để từ đó về sau, ta chỉ cần biết rằng chương trình của ta có một cấu trúc Stack, còn ta mô phỏng cụ thể như thế nào thì không cần phải quan tâm nữa, và khi cài đặt Stack bằng các cấu trúc dữ liệu khác, chỉ cần sửa lại các thủ tục StackInit, Push và Pop mà thôi.

**5.1.2. Mô tả Stack bằng danh sách liên kết đơn kiểu LIFO**

Khi cài đặt Stack bằng danh sách nối đơn kiểu LIFO, thì Stack bị tràn khi vùng không gian nhớ dùng cho các biến động không còn đủ để thêm một phần tử mới. Tuy nhiên, việc kiểm tra điều này rất khó bởi nó phụ thuộc vào máy tính và ngôn ngữ lập trình. Ví dụ như đối với Turbo Pascal, khi Heap còn trống 80 Bytes thì cũng chỉ đủ chỗ cho 10 biến, mỗi biến 6 Bytes mà thôi. Mặt khác, không gian bộ nhớ dùng cho các biến động thường rất lớn nên cài đặt dưới đây ta bỏ qua việc kiểm tra Stack tràn.

program StackByLinkedList;

type

PNode = ^TNode;

TNode = record;

Value: Integer;

Link: PNode;

end;

var

Last: Pnode;

procedure StackInit; {Khởi tạo Stack rỗng}

begin

Last := nil;

end;

procedure Push(V: Integer);

var

P: PNode;

begin

New(P); P^.Value := V;

P^.Link := Last; Last := P;

end;

function Pop: Integer;

var

P: PNode;

begin

if Last = nil then WriteLn('Stack is empty')

else

begin

Pop := Last^.Value;

P := Last^.Link;

Dispose(Last); Last := P;

end;

end;

begin

StackInit;

<Test>;

end.