CODELEARN

Mảng (Array) là một trong các cấu trúc dữ liệu cũ và quan trọng nhất. Mảng có thể lưu giữ một số phần tử cố định và các phần tử này nền có cùng kiểu. Hầu hết các cấu trúc dữ liệu đều sử dụng mảng để triển khai giải thuật. Dưới đây là các khái niệm quan trọng liên quan tới Mảng.

Phần tử: Mỗi mục được lưu giữ trong một mảng được gọi là một phần tử.

Chỉ mục (Index): Mỗi vị trí của một phần tử trong một mảng có một chỉ mục số được sử dụng để nhận diện phần tử.

Mảng gồm các bản ghi có kiểu giống nhau, có kích thước cố định, mỗi phần tử được xác định bởi chỉ số.

Mảng là cấu trúc dữ liệu được cấp phát liên tục cơ bản.

Ưu điểm của mảng :

Truy câp phần tử vơi thời gian hằng số O(1).

Sử dụng bộ nhớ hiệu quả.

Tính cục bộ về bộ nhớ.

Nhược điểm của mảng:

Không thể thay đổi kích thước của mảng khi chương trình dang thực hiện.

Mảng động:

Mảng động (dynamic aray) : cấp phát bộ nhớ cho mảng một cách động trong quá trình chạy chương trình trong C là malloc và calloc, trong C++ là new.

Sử dụng mảng động ta bắt đầu với mảng có 1 phần tử, khi số lượng phần tử vượt qua khả năng của ảng thì ta gấp đôi kích thước mảng cũ và copy phần tử mảng cũ vào nửa đầu của mảng mới.

Ưu điểm:

Tránh lãng phí bộ nhớ khi phải khai báo mảng có kích thước lớn ngay từ đầu.

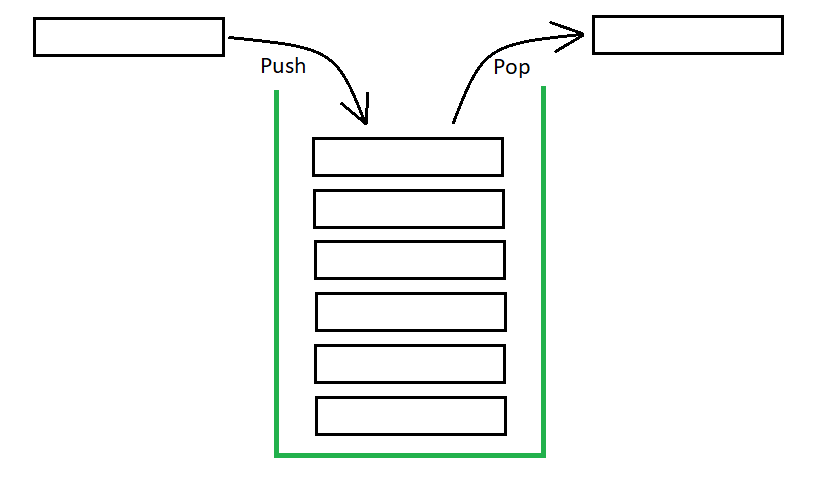
Nhược điểm:

Phải thực hiện them thao tác copy phần tử mỗi khi thay đổi kích thước.

Một số thời gian thực hiện thao tác không còn là hằng số nữa.

Giới thiệu về cấu trúc stack.

Stack là một loại container adaptor, được thiết kế để hoạt động theo kiểu LIFO (Last - in first - out) (vào sau ra trước), tức là một kiểu danh sách mà việc bổ sung và loại bỏ một phần tử được thực hiển ở cuối danh sách. Vị trí cuối cùng của stack gọi là đỉnh (top) của ngăn xếp.



Stack giống như việc giáo viên kiểm tra vở bài tập của học sinh vậy, ai nộp sau cùng thì vở bài tập của người đó sẽ được giáo viên kiểm tra đầu tiên, đương nhiên người nộp vợ đầu tiên sẽ được kiểm tra cuối cùng.

Stack có các hàm sau (ví dụ cho C++):

- size : trả về kích thước hiện tại của stack. ĐPT O(1).

- empty : true stack nếu rỗng, và ngược lại. ĐPT O(1).

- push : đẩy phần tử vào stack. ĐPT O(1).

- pop : loại bỏ phẩn tử ở đỉnh của stack. ĐPT O(1).

- top : truy cập tới phần tử ở đỉnh stack. ĐPT O(1)002E