**CÂU HỎI CHƯƠNG 2**

Câu 1:

* Trong khoa học máy tính, **danh sách đặc** được hiểu 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: Danh sách Họ và tên sinh viên có các phần tử thuộc cùng kiểu dữ liệu là kiểu chuỗi:

|  |  |
| --- | --- |
| Họ và tên |  |
| Nguyễn Văn A | 0 |
| Trần Văn B | 1 |
| Ngô Thị C | 2 |
| Võ Thị D | 3 |
| …. | … |
|  | n-1 |

Câu 2:

* Trong khoa học máy tính, **danh sách kiên kết** được hiểu là một danh sách mà các phần tử được cấp phát rời rạc nhau, và cố định trong bộ nhớ.
* Có 2 loại:
* Danh sách liên kết đơn: Mỗi phần tử trong danh sách gồm có 2 thành phần:
* Phần 1: vùng thông tin chứa giá trị cần quản lý
* Phần 2: vùng liên kết, chứa địa chỉ bộ nhớ của phần tử kế tiếp
* Danh sách liên kết kép: Mỗi phần tử trong danh sách gồm có 3 thành phần:
* Vùng chứa thông tin (info)
* Vùng liên kết (next) trỏ đến phần tử đứng liền sau nó
* Vùng liên kết (previous) trỏ đến phần tử đứng liền trước nó
* Ví dụ:

|  |  |  |
| --- | --- | --- |
| 0 | Lan | 1 |
| 1 | Chi | 3 |
| 2 | Hồng | NULL |
| 3 | Hoa | 2 |

* Danh sách liên kết đơn:

Hồng NULL

Hoa 2

Chi 3

Lan 1

* Danh sách liên kết kép:

3 Hoa 2

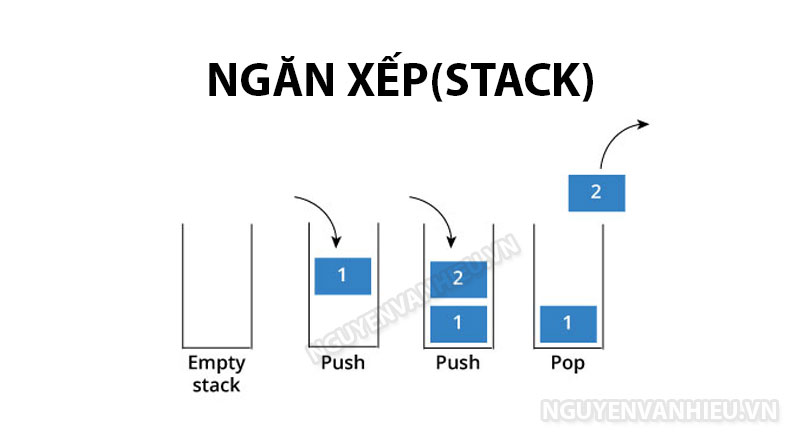
2 Hồng NULL

1 Chi 3

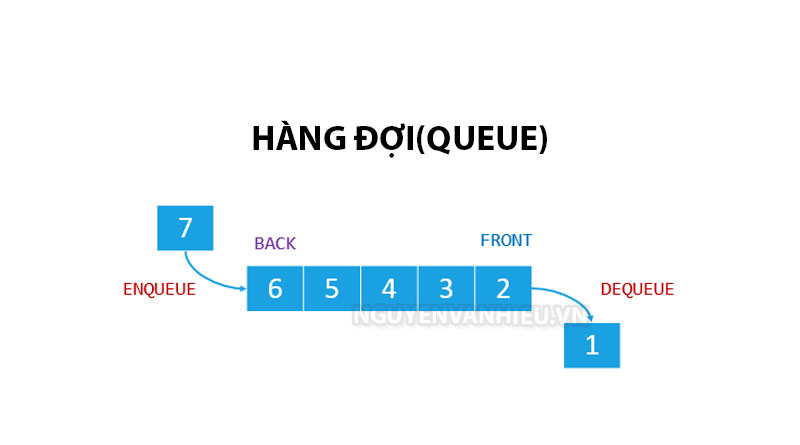
0 Lan 1

Câu 3:

* Nói STACK và QUEUE là danh sách hạn chế vì:
* Danh sách thường: có thể thêm, xóa phần tử ở vị trí bất kỳ trong danh sách.
* STACK (ngăn xếp): chỉ được thêm phần tử vào cuối danh sách, lấy ra phần tử cuối danh sách.
* QUEUE (hàng đợi): chỉ được thêm phần tử vào cuối danh sách, lấy ra phần tử ở đầu danh sách.
* Ví dụ:
* STACK:

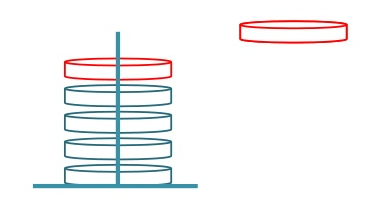


* QUEUE:

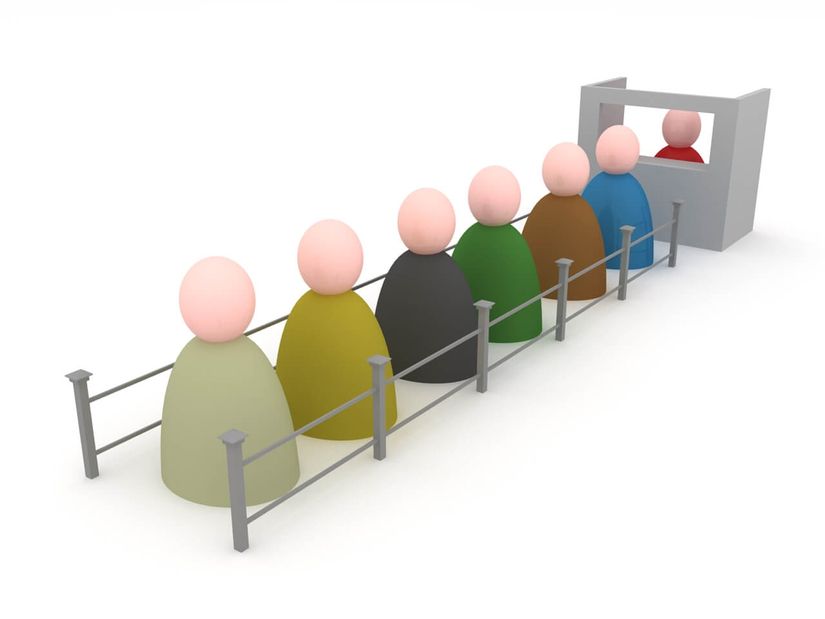


Câu 4:

* LIFO : Last In First Out: Cái được thêm vào sau thì sẽ được lấy ra trước
* FIFO: First In First Out: Cái được thêm vào trước sẽ được lấy ra trước
* Ví dụ:
* LIFO: vòng được thả vào cột sau thì được lấy ra trước



* FIFO: người vào trước thì được ra trước



Câu 6:

Danh sách liên kết có thể ứng dụng xử lý các vấn đề trong máy tính như:

* Tối ưu hóa bộ nhớ
* Cung cấp giải pháp để chứa cấu trúc dữ liệu tuyến tính
* Dễ dàng thêm hoặc xóa các phần tử trong danh sách mà không cần phải cấp phát hoặc tổ chức lại trật tự của mảng
* Cấp phát bộ nhớ động
* Xử lý các lệnh trong máy tính (ứng dụng trong HÐH, trình biên dịch)
* Bộ đệm (ví dụ: Nhấn phím -> Bộ đệm -> CPU xử lý)

Câu 7:

* Cấu trúc dữ liệu động là cấu trúc dữ liệu có thể kết hợp dữ liệu với những hình thức mới linh động hơn, có thể thay đổi kích thước, cấu trúc trong suốt thời gian thao tác.
* Ví dụ: Kiểu dữ liệu con trỏ, Danh sách liên kết