

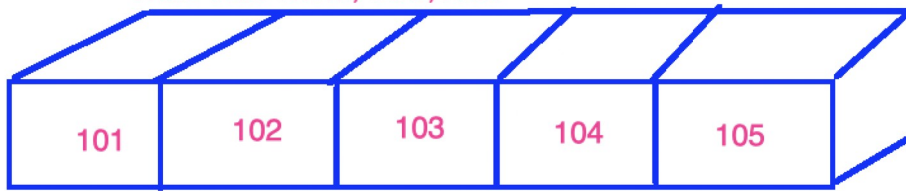
## Slot14- Data Structure

Mảng 1 chiều

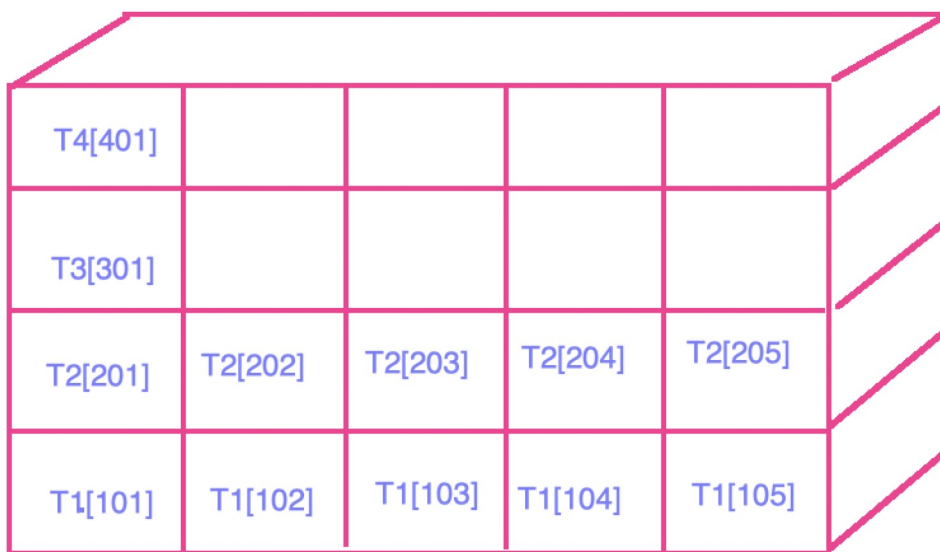
Mỗi 1 phòng học= 1 phần tử của mảng tầng 1  
Thao Tác: Thêm, Sửa, Xóa

Mỗi phần tử có:

- Địa chỉ: T1[101]
- Giá trị: T1[101]=30



1 vòng lặp



2 vòng lặp

----

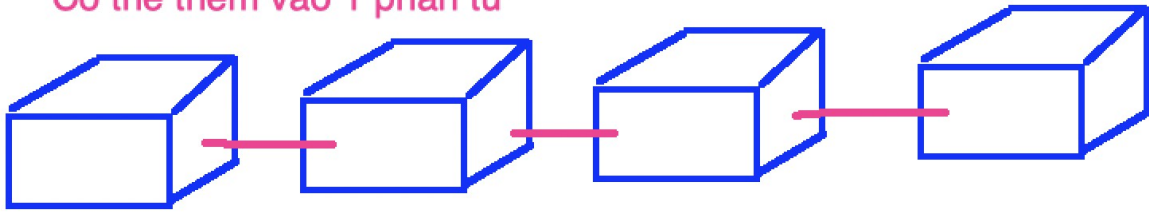
## Record

	Ma SV	TenSV	Diem
Record1	HENVA	Nguyễn Van A	8
Record2	HETVB	TRan Van B	9
Record3	HEVVC	Vu Van C	7
Record4	HENTD	Nguyễn Thi D	8.5

-----

Linked List

Có thể thêm vào 1 phần tử

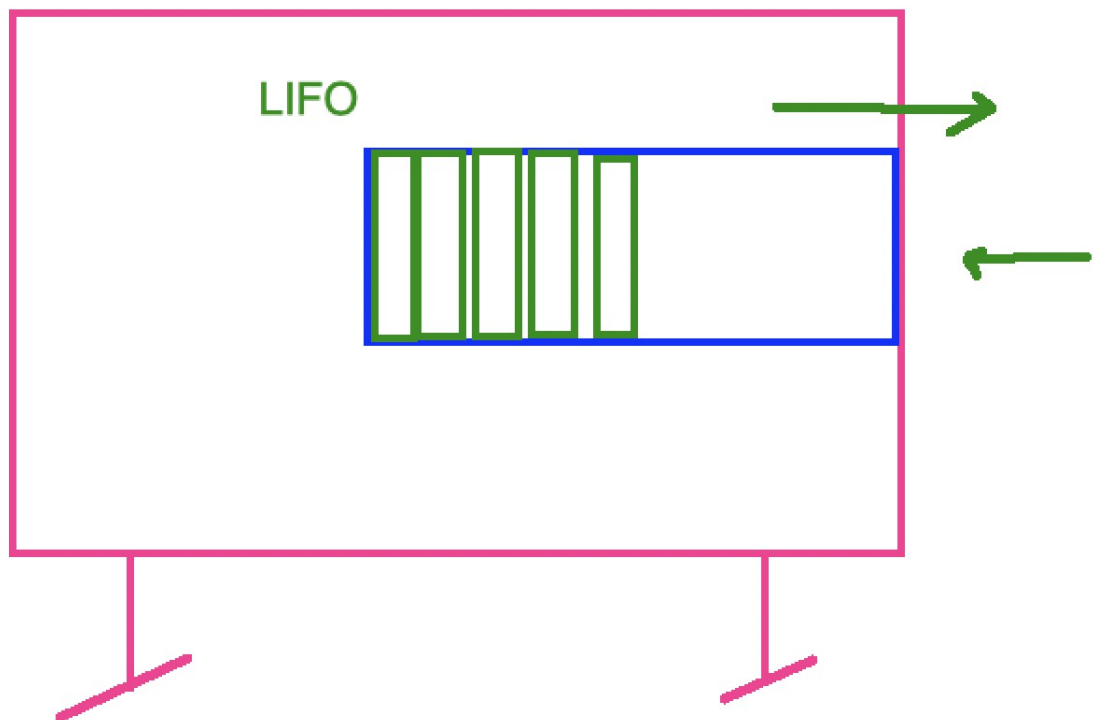


Áp dụng khi ta chưa biết sẽ lưu bao nhiêu dữ liệu

-----

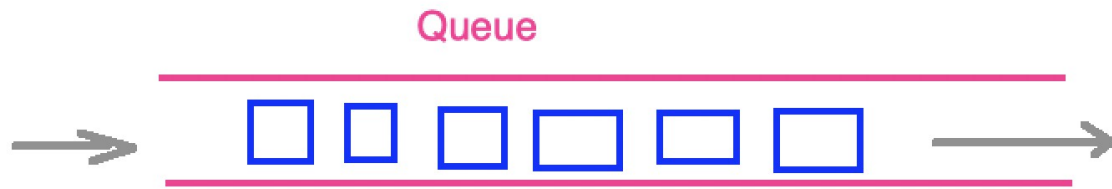
Stack: ngăn xếp

Stack



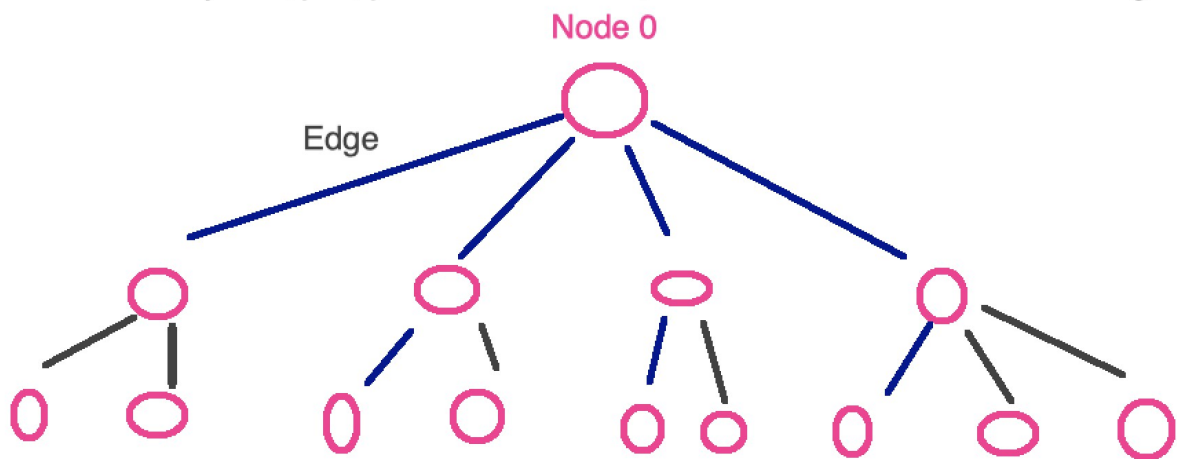
Xếp hàng mua cơm ở Càng tin: Hàng đợi (Queue)

Xếp trước -> Mua trước: FIFO



----

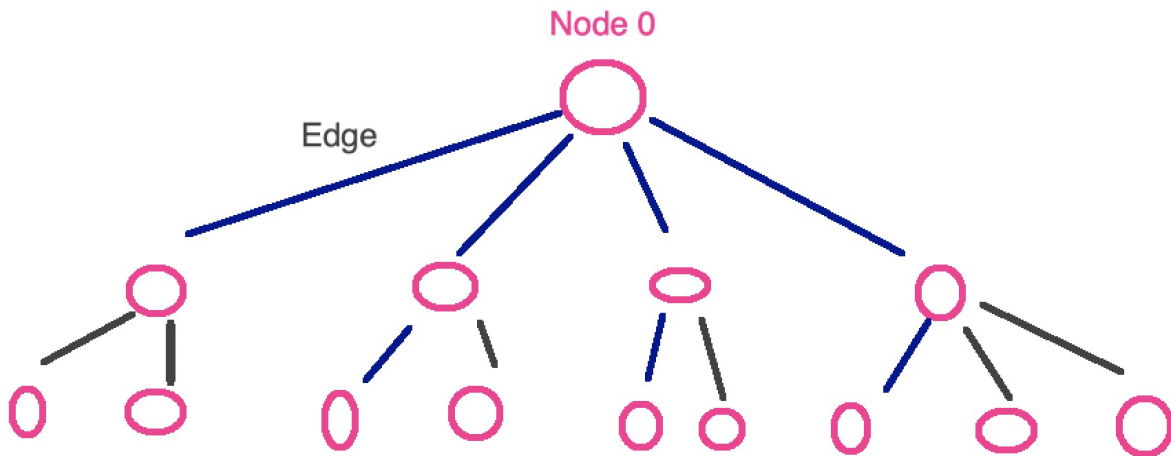
Cây: là tập hợp các nút và các cạnh kết nối với nhau. Nút 0 là nút gốc



---

Cây nhị phân:

Cây: là tập hợp các nút và các cạnh kết nối với nhau. Nút 0 là nút gốc



Cây nhị phân:

Mỗi cây có không quá 2 cây con

Các thao tác: Duyệt theo chiều sâu (các đời: cụ-> ông-> bố->tôi-> con)

Các thao tác: Duyệt theo chiều rộng: các anh em

Ứng dụng trong Mã hóa

----

Mảng:

Tính tổng các phần tử trong mảng

Tìm giá trị lớn nhất trong mảng

Bản ghi:

Quản lý thông tin sinh viên

```
#include <stdio.h>
int main(){
    int arr[5]={1,3,4,5,6}; //khai bao mang
    int sum=0; // gan tong = 0
    for(int i=0;i<5;i++){
        sum += arr[i]; //cong don vao tong
    }
    printf("Tong cac phan tu trong mang la: %d\n",sum);
    return 0;
}
```

----

```
#include <stdio.h>
int main(){
    int arr[5]={1,3,120,5,6}; // khai bao mang
    int max=arr[0]; // gan tong = phan tu dau tien
    for(int i=0;i<5;i++){
        if(arr[i]>max){
            max=arr[i];
        }
    }
    printf("Max la: %d\n",max);
    return 0;
}
```

----

```

#include <stdio.h>
#include <stdlib.h>
//dinh nghia cau truc ban ghi sinh vien
struct Student{
    int id;
    char name[50];
    float grade;
};
int main(){
    int n;
    //nhap vao so luong sinh vien
    printf("Nhap vao so luong sinh vien: ");
    scanf("%d",&n);
    //xin cap phat bo nho dong cho danh sach sinh vien
    struct Student* students=(struct Student*)malloc(n*sizeof(struct Student));
    //nhap thong tin cua tungf sinh vien
    for(int i=0;i<n;i++){
        printf("\n Nhap thong tin cho sinh vien thu %d\n ",i+1);
        printf("Nhap ID: ");
        scanf("%d",&students[i].id);//nhap id
        printf("Nhap ten: ");
        scanf(" %[^\\n]s",students[i].name);//nhap name
        printf("Nhap diem: ");
        scanf("%f",&students[i].grade);//nhap diem
    }

    //sau khi nhap xong ta can hien thi thong tin cua tat ca cac sinh vien
    printf("\nThong tin cac sinh vien da nhap: \n");
    for(int i=0;i<n;i++){
        printf("\nSinh vien tghu %d\n",i+1);
        printf("ID: %d\n",students[i].id);//in ra id
        printf("Name: %s\n",students[i].name);//in ra id
        printf("Grade: %.2f\n",students[i].grade);//in ra id
    }
    //giai phong bo nho
    free(students);
    return 0;

```