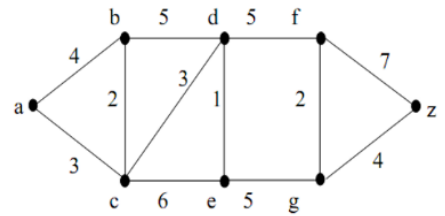


## Mảng L và mảng trước qua các lần lặp



### Mảng L[] và mảng trước[] ở lần khởi tạo

	A	B	C	D	E	F	G	Z
Khởi tạo	0*	vc	vc	vc	vc	vc	vc	vc

Giá trị	0	0	0	0	0	0	0	0
Truoc	A	B	C	D	E	F	G	Z

### Mảng L[] và mảng trước[] sau lần lặp 1

	A	B	C	D	E	F	G	Z
Lặp 1	0	4	3*	vc	vc	vc	vc	vc

Giá trị	0	1	1	0	0	0	0	0
Truoc	A	B	C	D	E	F	G	Z

### Mảng L ở lần lặp 2

	A	B	C	D	E	F	G	Z
Khởi tạo	0*	vc	vc	vc	vc	vc	vc	vc
Lặp 1	0	4	3*	vc	vc	vc	vc	vc
Lặp 2	0	4	3	6	9	vc	vc	vc

### Mảng trước[] ở lần lặp 2

Giá trị	0	1	1	3	3	0	0	0
Truoc	A	B	C	D	E	F	G	Z

v.v...

### Mảng L ở lần lặp 8

	A	B	C	D	E	F	G	Z
Khởi tạo	0*	vc	vc	vc	vc	vc	vc	vc
Lặp 1	0	4	3*	vc	vc	vc	vc	vc
Lặp 2	0	4	3	6	9	vc	vc	vc
...								
Lặp 8	0	4	3	6	7	11	12	16

### Mảng trước[] ở lần lặp 8

Giá trị	0	1	1	3	4	4	5	7
Truoc	A	B	C	D	E	F	G	Z

## Các công việc

### 1. Bước khởi tạo

viết hàm khoitao(int s) với s là đỉnh xuất phát, có thể tận dụng hàm này để đánh dấu các đỉnh chưa duyệt (tương tự bài dfs) luôn. Dùng lệnh #define vocung 1000 để định nghĩa biến vô cùng ở phần khai báo.

**Khoitao(int s)** //khởi tạo nhãn cho các đỉnh (mảng L[]) và mảng đánh dấu (d[])

```
{
    {
        L(vi) = vocung
        d(vi) = 1
    }
    L(s)=0
}
```

2. Viết hàm trả về **vị trí có giá trị nhỏ nhất** trong mảng một chiều L

**vitridinhmin()** //tìm vị trí (đỉnh) có giá trị nhỏ nhất trong mảng L

```
{
    // viết hàm tìm vị trí min trong mảng như đã học ở kỹ thuật lập trình
    // lưu ý: đỉnh min được tìm thấy phải là đỉnh chưa đánh dấu
}
```

3. Viết hàm cập nhật nhãn các đỉnh kề của đỉnh có nhãn nhỏ nhất

**capnhatke\_p(int p)** //tính lại nhãn các đỉnh kề của đỉnh p

```
{
    for tất cả các đỉnh i=1...n
        nếu đỉnh i kề với đỉnh p và i chưa được đánh dấu
            nếu L(i) > L(p) + a(p,i)
                thì L(i) = L(p) + a(p,i)
            truoc[?]=?
}
```

4. Viết hàm tìm đường đi từ đỉnh bắt đầu s đến đỉnh kết thúc f dựa vào mảng trước

**timduong(int f)** //tim duong tu dinh s den dinh f

```
{
    Độ dài đường đi từ s đến f là: L[f]
    //tìm các đỉnh đi qua
    Khi truoc[y] còn khác 0 thì
    {
        In giá trị truoc(y)
        Gán y=truoc(y)
    }
}
```

5. Viết hàm dijkstra tổng hợp lại các hàm trên

**Dijkstra()**

```
{
    Nhập đỉnh xuất phát s
    Khoitao(s) (hàm 1)
    Khi lặp != n lần
    {
        Tìm u là đỉnh có giá trị nhãn nhỏ nhất và chưa đánh dấu (hàm 2)
        Cập nhật u (hàm 3)
        Đánh dấu u
        Tăng biến lặp
    }
    Nhập đỉnh kết thúc y
    Tìm đường đi ngắn nhất giữa đỉnh s và y (hàm 4)
}
```