**Báo Cáo Thực Hành Buổi 1**

**Trần Minh Trí**

**DTH185413**

[1. Giải Thuật Breath First Search (BFS): 2](#_Toc66032080)

[2. Giải Thuật Depth First Search (DFS) 2](#_Toc66032081)

[3. Giải Thuật Best-FirstSearch (Best-FS) 2](#_Toc66032082)

[4. Giải thuật A Start (A\*): 3](#_Toc66032083)

# Giải Thuật Breath First Search (BFS):

## Dữ liệu đầu vào của chương trình là ma trận kề và cặp đỉnh kề.

## Tổ chức dữ liệu:

### Open là Queue.

### Close là visit.

### Cost là top và bottom.

## Cài đặt giải thuật

### Cài đặt lấy phần tử trong Open: đọc file đưa vào mảng 2 chiều a[i][j]

### Kiểm tra xem có đỉnh cần tìm: Nếu Phần tử tại vị trí đang xét trong mảng khác 0 và chưa xét thì nhận biết là đỉnh

### Phát sinh các con:

#### Không nằm trong Open, Close: nó không nằm trong vùng chờ xét cũng không nằm trong vùng đã xét

#### Nằm trong Close: là chứa các phần tử đã xét

#### Đưa vào open: chứa các phần tử trong vùng chờ xét

#### Đưa vào open: phần tử được đưa vào khi phần tử đó khác 0 và phần tử đó chưa được xét và ta sẽ lấy phần tử đó đưa vào cuối hang đợi và gán nhãn cho nó là đã duyệt.

# Giải Thuật Depth First Search (DFS)

## Dữ liệu đầu vào là ma trận kề.

## Tổ chức dữ liệu:

### Open: là Stack.

### Close: là chứa các đỉnh đã xét

### Cost: biểu diễn visit các đỉnh được xét với 2 giá trị là True và False, với True là đã xét còn false là chưa xét

## Cài đặt giải thuật:

### Cài đặt lấy phần tử trong Open: đọc file đưa vào mảng 2 chiều a[i][j].

### Kiểm tra xem có đỉnh cần tìm: Nếu Phần tử tại vị trí đang xét trong mảng bằng 1 và chưa xét thì nhận biết là đỉnh.

### Phát sinh các con:

#### Không năm trong open , close: sẽ không được thực hiện

#### Nằm torng open: sẽ được thực hiện ở vòng lập tiếp theo

#### Đưa vào Open: nếu thỏa điều kiện kề với đỉnh đang xét và chưa xét trước đó sẽ them vào Open.

# Giải Thuật Best-FirstSearch (Best-FS)

## Dữ liệu đầu vào là Cặp Đỉnh Kề có trọng số vã hẻuistic.

## Tổ chức dữ liệu:

### Open: Open

### Close:Close

### Cost: lưu vào cost dưới dạng ma trận và tính chi phí. Trước khi gán, cho cost[i][j]=0

## Cài Đặt Giải Thuật

### Cài đặt lấy phần tử trong Open: Thiết lập nút bắt đầu và thêm vào open

### Kiểm tra đỉnh cần tìm: lấy nút nhỏ hơn 0 , để bắt đầu vòng lặp mới

### Phát sinh các con:

#### Không nằm trong Open, Close: thiết lập giá trị và lưu vào open

#### Nằm trong Open: Nếu đường đi ngắn nhất thì Display đỉnh duyệt, lấy ra khỏi Open và thêm vào Close, nếu tìm được đường đi tốt hơn thì lấy ra giá trị cũ ra khỏi Open, thay bằng giá trị tốt hơn và thêm lại giá trị đó vào Open

#### Nằm trong Close: sẽ được bỏ qua mà không thực hiện.

#### Đưa vào Open: nếu thỏa điều kiện với các đỉnh đang xét.

### Danh sách trả về lộ trình có tổng trọng số nhỏ nhất.

# Giải thuật A Start (A\*):

## Dữ liệu đầu vào là dữ liệu tập tin crow.txt và Romania.text

## Tổ chức dữ liệu:

### Open: Trạng thái chưa xét

### Close: tập trạng thái chưa xét.

### Cost: là khoảng cách giữa trạng thái đầu và trạng thái hiện tại, giữa trạng thái hiện tại và trạng thái đích , và lưu dưới dạng ma trận kề

## Cài Đặt Giải Thuật

### Cài đặt lấy phần tử trong Open: Xét các trạng thái tại tốt nhất trong Open và lưu nó vào Close, nếu nó là trạng thái đích thì kết thúc, tạo ra các trạng thái còn lại trong Open và đưa vào Close, nếu tìm được đường đi tốt hơn thì lấy giá trị cũ ra khỏi Open, thay bằng giá trị tốt hơn và thêm lại giá trị đó vào Open

### Kiểm tra đỉnh cần tìm: lấy nút nhỏ hơn 0 , để bắt đầu vòng lặp mới

### Phát sinh các con:

#### Không nằm trong Open, Close: thiết lập giá trị và lưu vào open

#### Nằm trong Open: Nếu đường đi ngắn nhất thì Display đỉnh duyệt, lấy ra khỏi Open và thêm vào Close, nếu tìm được đường đi tốt hơn thì lấy ra giá trị cũ ra khỏi Open, thay bằng giá trị tốt hơn và thêm lại giá trị đó vào Open

#### Nằm trong Close: sẽ được bỏ qua mà không thực hiện.

#### Đưa vào Open: Nếu không nằm trong Open và Close thì thiết lập danh sách đưa vào Open.