




Hướng dẫn thực hành BFS, DFS




## MỤC TIÊU, TÓM TẮT

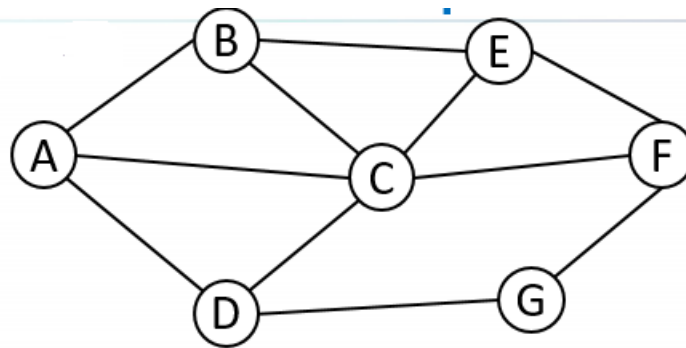
Khái niệm:

### THUẬT GIẢI BFS

-  **Bước 0:** Giả sử **s** là đỉnh **bắt đầu**, cho **s** vào trong Q (Queue)
-  **Bước 1:** Nếu Q khác **rỗng**, gọi **u** là phần tử được lấy từ Q, và xuất **u** ra ngoài màn hình;
-  **Bước 2:** Tìm các đỉnh **v** kề với **u**, cho **v** vào trong Q quay lại bước 1.

### Ý TƯỞNG CÀI ĐẶT

-  Dùng một danh sách đặc (mảng) **bfs** để lưu tập đỉnh đã duyệt.
-  Dùng tập Q (**Queue**) để lưu các đỉnh kề với đỉnh vừa duyệt.
-  Dùng tập C (**chuaxet**) để lưu các đỉnh chưa được duyệt (mới bắt đầu).



Kết Quả duyệt theo BFS:

A B C D E F G

Sinh viên tự viết và nhập ma trận kề theo hướng dẫn buổi 9

.....

## Ý TƯỞNG DFS

- 📖 Bắt đầu từ một đỉnh **u**
- 📖 Từ **u** đi theo cạnh (cung) xa nhất có thể
- 📖 Nếu hết đường đi, trở lại đỉnh trước của cạnh xa nhất tiếp tục duyệt như trước, cho đến đỉnh cuối cùng.

## THUẬT GIẢI DFS

- 📖 **Bước 0:** Giả sử **v** là đỉnh bắt đầu, cho **v** vào Stack và cho **v** vào tập DFS;
- 📖 **Bước 1:** Nếu Stack chưa rỗng, thì lấy **x** từ trong Stack ra;
- 📖 **Bước 2:** Tìm **u** (gần nhất) kề với **x** và **u** chưa có trong tập DFS;
  - 2.1. cho **x** vào trong Stack,
  - 2.2. **u** vào trong Stack;
  - 2.3. Cho **u** vào tập DFS;
  - 2.4. Sau đó quay lại bước 1;

## Ý TƯỞNG CÀI ĐẶT

- 📖 Dùng một danh sách **dfs** để lưu tập đỉnh đã duyệt.
- 📖 Dùng sp (**Stack**) để lưu các đỉnh kề với đỉnh vừa duyệt.
- 📖 Dùng tập **C** (**chua xet**) để lưu các đỉnh chưa được duyệt (lúc mới bắt đầu).

## CHƯƠNG TRÌNH MẪU

```
//          Duyet BFS (QUEUE)
int C[100], bfs[100];
int nbfs=0;
void InitC()
{
    for(int i=0; i<n; i++) // n la so dinh cua do thi
        C[i]=1;
}
void BFS(int v) // v la dinh bat dau
{
    int w, p;
    PushQ(v);
    C[v]=0;
    while(front!=NULL)
    {
        PopQ(p);
        bfs[nbfs]=p;
        nbfs++;
    }
}
```

```
    for(w=0; w<n; w++)
        if(C[w] && A[p][w]==1)
        {
            PushQ(w);
            C[w]=0;
        }
}
```

```
//          Duyet DFS (STACK)
int dfs[100];
int ndfs=0;
void DFS(int s)
{
    PushS(s);
    dfs[ndfs]=s;
    ndfs++;
    C[s]=0;
}
```

```

int v=-1, u=s;
while(isEmptyS()==0)
{
    if(v==n)
        PopS(u);
    for(v=0;v<n;v++)
        if(A[u][v]!=0 && C[v]==1)
        {
            PushS(u);
            PushS(v);
            dfs[ndfs]=v;
            ndfs++;
            C[v] = 0;
            u=v;
            break;
        }
}
}

```

.....

// SV TỰ BỔ SUNG ĐỂ CHO RA MÀN HÌNH NHƯ YÊU CẦU BÊN DƯỚI

## Bài 1:

Làm như hướng dẫn bên trên và xuất ra được màn hình như bên dưới

```

1. ----- BAI TAP 4 , CHUONG 5 , DO THI -----
2. 1. Khoi tao MA TRAN KE rong
3. 2. Nhap MA TRAN KE tu file text
4. 3. Nhap MA TRAN KE
5. 4. Xuat MA TRAN KE
6. 5. Duyet do thi theo chieu rong BFS - MTK
7. 6. Duyet do thi theo chieu sau DFS - MTK
8. 7. Tim dinh co gia tri x theo BFS
9. 8. Thoat

10. Vui long chon so de thuc hien: 2
11. Ban vua nhap MA TRAN KE tu file:
12. 0 1 1 1 0 0 0
13. 1 0 1 0 1 0 0
14. 1 1 0 1 1 1 0
15. 1 0 1 0 0 0 1
16. 0 1 1 0 0 1 0
17. 0 0 1 0 1 0 1
18. 0 0 0 1 0 1 0

19. Vui long chon so de thuc hien: 5
20. Vui long nhap dinh xuất phát: 0
21. Thu tu dinh sau khi duyệt BFS:
22. A B C D E F G

```

Làm tương tự cho thuật toán DFS, chọn số 6 trong menu và thực thi.

.....