

Chuong Le Hoang

Open University

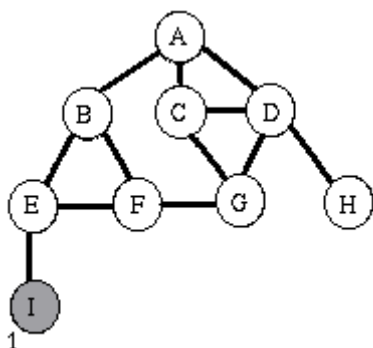
Tháng Mười Một 24, 2013

Thuật toán DFS – Tìm kiếm theo chiều sâu

2 phản hồi

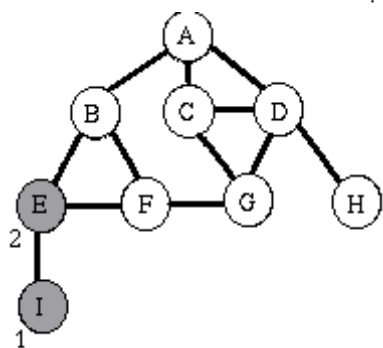
1. Mô tả:

- Đây là thuật toán tìm các đỉnh bằng cách duyệt theo chiều sâu.
- Xuất phát từ 1 đỉnh và đi mãi cho đến khi không thể đi tiếp, sau đó đi về lại đỉnh đầu. Trong quá trình quay lại:
 - + nếu gặp đường đi khác thì đi cho đến khi không đi tiếp được nữa
 - + nếu không tìm ra đường đi nào khác thì ngừng việc tìm kiếm.
- Trong quá trình đi đến đỉnh khác, thuật toán sẽ lưu lại đỉnh cha vừa đi qua để khi đi ngược lại từ đỉnh Kết thúc đến đỉnh Xuất phát, ta có thể xem được đường đi từ đỉnh Kết thúc đến đỉnh Bắt Đầu (có thể số lần đi không ít nhất, các bạn có thể tham khảo thuật toán **BFS** (<https://lhchuong.wordpress.com/2013/11/08/thuat-toan-bfs-tim-kiem-theo-chieu-ro%CC%A3ng/l>)).
- Sở dĩ thuật toán này tìm được đường đi là nhờ vào cơ chế tô màu và lưu đỉnh cha. Quá trình tô màu khiến 1 đỉnh không thể xét 2 lần trở lên và có thể xem được đường đi từ đỉnh Kết Thúc đến đỉnh Xuất phát dựa vào việc lưu đỉnh cha.
- Sau đây là minh họa về thuật toán:
 - + **Hình 1** đi từ đỉnh bắt đầu, đi cho đến khi không đi được nữa.



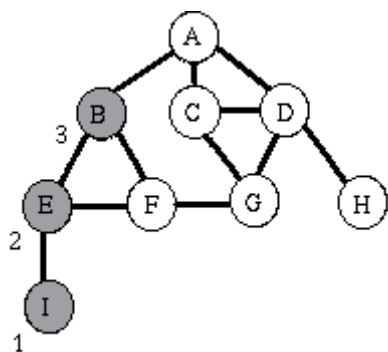
(<https://lhchuong.files.wordpress.com/2013/11/1.png>)

Hình 1



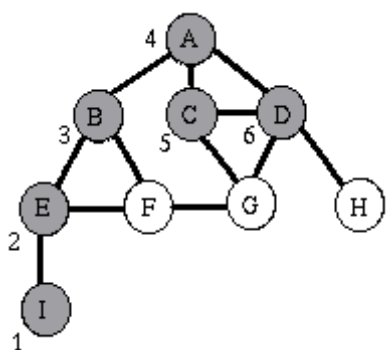
<https://lhchuong.files.wordpress.com/2013/11/21.png>

Hình 2



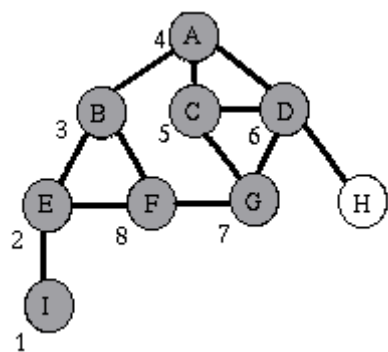
<https://lhchuong.files.wordpress.com/2013/11/3.png>

Hình 3



<https://lhchuong.files.wordpress.com/2013/11/4.png>

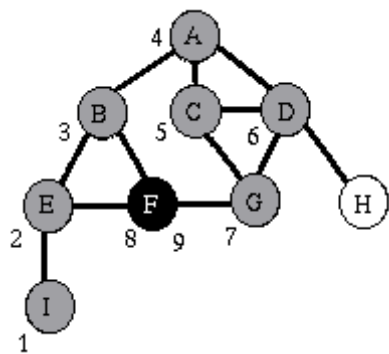
Hình 4



<https://lhchuong.files.wordpress.com/2013/11/5.png>

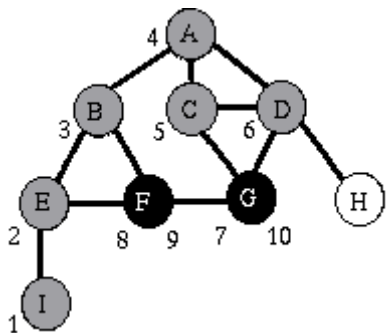
Hình 5

+ **Hình 6** do không đi được nữa nên quay ngược về lại đỉnh bắt đầu.



<https://lhchuong.files.wordpress.com/2013/11/61.png>

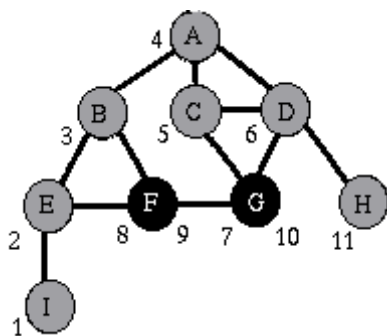
Hình 6



<https://lhchuong.files.wordpress.com/2013/11/71.png>

Hình 7

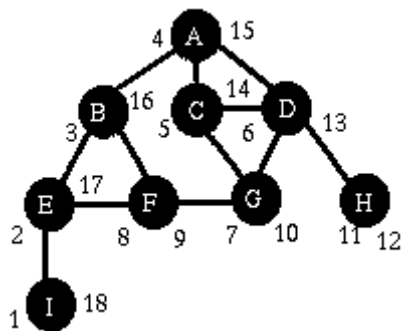
+ **Hình 8** khi quay lại đến đỉnh D, gặp đỉnh H vẫn chưa được tô màu, tìm được đường đi mới.



<https://lhchuong.files.wordpress.com/2013/11/8.png>

Hình 8

+ **Hình 9** sau khi đi qua đỉnh H, không thể đi tiếp được nữa nên tiến hành quay lại đến đỉnh xuất phát.



<https://lhchuong.files.wordpress.com/2013/11/9.png>

Hình 9

2. Cài đặt:

```
//khai báo mảng phần tử
int A[MAX][MAX] = {
    {0,2,1,0,0,3},
    {2,0,0,0,0,5},
    {1,0,0,1,3,0},
    {0,0,1,0,2,0},
    {0,0,3,2,0,3},
    {3,5,0,0,3,0}
};
int V = 6;
//khai báo các mảng dữ liệu
int TIME = 0;
int color[MAX], times[MAX], back[MAX];
```

(<https://lhchuong.files.wordpress.com/2013/11/11.png>)

Hình 10

– **Hình 11:** khởi tạo các mảng và duyệt từng đỉnh theo chiều sâu

```
void DFS()
{
    //khởi tạo giá trị ban đầu của các mảng
    for(int i = 0; i < V; i++)
    {
        color[i] = 0;
        back[i] = times[i] = -1;
    }

    //duyet từng đỉnh
    for(int i = 0; i < V; i++)
        if(color[i] == 0)
            Visit(i);
```

(<https://lhchuong.files.wordpress.com/2013/11/22.png>)

Hình 11

– **Hình 12:** Visit là hàm duyệt theo chiều sâu, với tham số **u** là đỉnh sẽ duyệt

```

void Visit(int u)
{
    //đánh dấu đỉnh vừa được đi qua
    color[u] = 1;
    TIME++;
    //lưu lại số lần duyệt
    times[u] = TIME;
    //duyệt các đỉnh kề
    for(int v = 0; v < V; v++)
        if(A[u][v] != 0 && color[v] == 0)
        {
            back[v] = u;
            times[v] = times[u] + 1;
            //đi tới đỉnh kế tiếp
            Visit(v);
        }
    //sau khi đi xong thì quay ngược lại đỉnh xuất phát
    color[u] = 2;
}

```

(<https://lhchuong.files.wordpress.com/2013/11/31.png>)

Hình 12

– *Hình 13:* Xem kết quả

```

//kiểm tra kết quả cài đặt
cout << "Nhập đỉnh bắt đầu: ";
int start, finish;
cin >> start;
cout << "Nhập đỉnh kết thúc: ";
cin >> finish;
//thực hiện in đường đi từ start -> kết thúc
PrintPath(start, finish);

```

(<https://lhchuong.files.wordpress.com/2013/11/41.png>)

Hình 13

– *Hình 14:* In đường đi từ đỉnh Xuất phát đến đỉnh kết thúc dựa vào mảng **Back**

```
void PrintPath(int start, int finish)
{
    if(start == finish)
        cout << finish << " ";
    else
        if(back[finish] == -1)
            cout << "Khong co duong di\n";
        else
        {
            PrintPath(start, back[finish]);
            cout << finish << " ";
        }
}
```

<https://lhchuong.files.wordpress.com/2013/11/51.png>

Hình 14

Chúc các bạn thành công ^_^

Advertisements

[Report this ad](#)

[Report this ad](#)

Posted by [Chuong Le Hoang](#) in [Data structures & Algorithms](#)

2 thoughts on “Thuật toán DFS – Tìm kiếm theo chiều sâu”

1. **td** nói:

Tháng Bảy 19, 2015 lúc 7:31 chiều

Xin chào,

Mình có 1 thắc mắc như sau:

Theo code của bạn ở hình 12 thì khi đến một nút mà ko có con, chương trình sẽ ko thể quay ngược lại nút cha của nút con đó -> như vậy sẽ có thể sẽ bỏ qua các nút còn lại chưa được duyệt, ý mình muốn nói ứng với hình 6 ở trên: từ F (ko có con) ko quay được về G -> ko quay được về D -> ko tìm ra H ?

Bạn có tô màu các nút bằng mảng color với các giá trị 0, 1, 2, tuy nhiên chỉ dùng giá trị 0 để kiểm tra nút đã được thăm hay chưa. Mình nghĩ quá trình 1 nút con quay ngược lại 1 nút cha cần sử dụng cả giá trị 1 và 2 nữa.

Mong sớm nhận được hồi đáp của bạn.

Thân.

Phản hồi

o **Minh Hien** nói:

Tháng Mười Một 16, 2017 lúc 10:08 chiều

Có đoạn for $i=1; i \leq v$ ở hình 11 đấy bạn

Phản hồi

[Blog tại WordPress.com.](https://lchuong.wordpress.com/2013/11/24/thua%CC%A3t-toan-dfs-tim-kiem-theo-chieu-sau/)