

# Chuong Le Hoang

## Open University

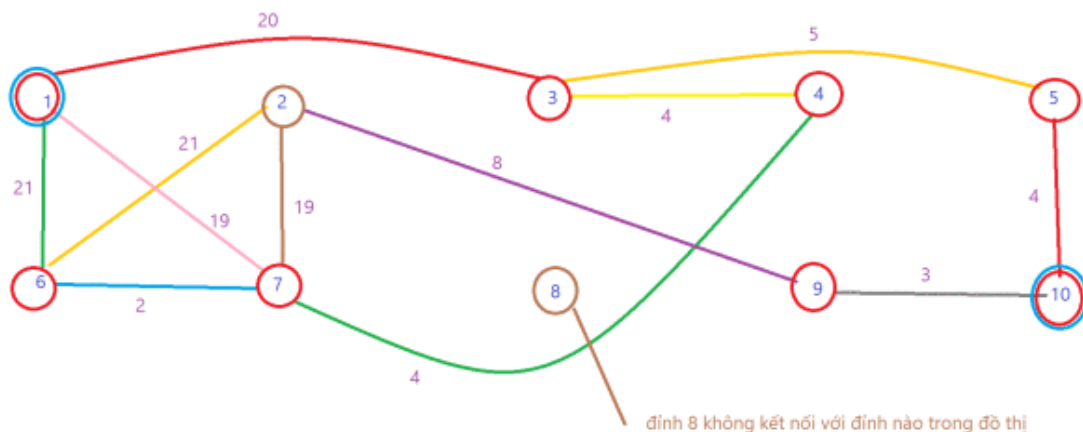
Tháng Chín 6, 2013

## Thuật toán Dijkstra – Tìm đường đi ngắn nhất dựa vào trọng số

### 5 phản hồi

#### 1. Mô tả:

– Đồ thị sẽ được tổ chức như *Hình 1*:



<https://lhchuong.files.wordpress.com/2013/09/graph.png>

**Hình 1**

– Chúng ta sẽ thực hiện việc tìm đường đi ngắn nhất dựa vào trọng số từ đỉnh 1 -> đỉnh 10.

#### 2. Cài đặt:

Chúng ta sẽ tiến hành cài đặt bằng ngôn ngữ C++

– *Hình 2*: tiến hành lưu đồ thị trên theo ma trận kề, với 10 đỉnh

```
#include <iostream.h>
#include <limits.h>

int mat[100][100] = {
    {0, 0, 20, 0, 0, 21, 19, 0, 0, 0},
    {0, 0, 0, 0, 0, 21, 19, 0, 8, 0},
    {20, 0, 0, 4, 5, 0, 0, 0, 0, 0},
    {0, 0, 4, 0, 0, 0, 4, 0, 0, 0},
    {0, 0, 5, 0, 0, 0, 0, 0, 0, 4},
    {21, 21, 0, 0, 0, 0, 2, 0, 0, 0},
    {19, 19, 0, 4, 0, 2, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 8, 0, 0, 0, 0, 0, 0, 0, 3},
    {0, 0, 0, 0, 4, 0, 0, 0, 3, 0}
};
int n = 10;
```

(<https://lhchuong.files.wordpress.com/2013/09/untitled.png>)

## Hình 2

– Hình 3: Gọi hàm trong **main**, ở đây 0 là đỉnh bắt đầu và 9 là đỉnh kết thúc

```
void main()
{
    dijkstra(0,9);
}
```

(<https://lhchuong.files.wordpress.com/2013/09/untitled1.png>)

## Hình 3

– Hình 4: Hàm **dijkstra**

+ **input**: đỉnh bắt đầu và đỉnh kết thúc

+ **output**: tổng trọng số và thứ tự đi giữa các đỉnh từ đỉnh bắt đầu đến đỉnh kết thúc.

```

void dijkstra(int start, int finish)
{
    int back[100], //luu dinh cha
        weight[100], //luu trọng số
        mark[100]; //đánh dấu đỉnh
    //khởi tạo
    for(int i = 0; i < 10; i++)
    {
        back[i] = -1;
        mark[i] = 0;
        weight[i] = INT_MAX;
    }

    //xuất phát tại đỉnh đầu tiên
    back[start] = 0;
    weight[start] = 0;
    //kiểm tra đồ thị có liên thông hay k
    int connect;
    do
    {
        //đỉnh bắt đầu sẽ là đỉnh 0
        //nên ta gán connect = -1
        connect = -1;
        int min = INT_MAX;
        //lần lượt duyệt qua tất cả các đỉnh trong đồ thị
        for(int j = 0; j < n; j++)
            if(mark[j] == 0) //nếu đỉnh chưa được đánh dấu
            {
                //nếu tồn tại đường đi giữa đỉnh start và đỉnh j
                //weight[j]: tổng trọng số từ đỉnh bắt đầu đến đỉnh đang xét
                //weight[start] + mat[start][j]: trọng số đang xét
                if(mat[start][j] != 0 &&
                    weight[j] > weight[start] + mat[start][j])
                {
                    //luu lại, dùng để so sánh lần sau
                    weight[j] = weight[start] + mat[start][j];
                    //luu dinh cha
                    back[j] = start;
                }

                //dựa vào mảng weight -> tìm đường đi ngắn nhất hiện tại
                if(min > weight[j])
                {
                    min = weight[j];
                    //dựa vào biến connect,
                    //ta có thể quyết định được đỉnh tiếp theo cần duyệt
                    //và đồ thị có liên thông không
                    connect = j;
                }
            }
        start = connect;
        mark[start] = 1;
    } while(connect != -1 && start != finish);
    //start != finish: đỉnh đầu và đỉnh cuối gặp nhau
    //connect != -1: nếu không liên thông thì dừng việc tìm đường đi ngắn nhất

    //xem trọng số vừa tìm được
    cout << weight[finish] << endl;
    //in đường đi
    printPath(0, finish, back);
    cout << "null\n";
}

```

(<https://lhchuong.files.wordpress.com/2013/09/untitled5.png>)

Hình 4

– Hình 5: Hàm **printPath**

+ **input**: đỉnh bắt đầu và đỉnh kết thúc, mảng lưu thứ tự các đỉnh

+ **output**: in đường đi đã được thiết lập từ đỉnh bắt đầu và đỉnh kết thúc

```
void printPath(int start,int finish, int back[])
{
    if(start == finish)
        cout << finish << " -> ";
    else
    {
        printPath(start,back[finish],back);
        cout << finish << " -> ";
    }
}
```

<https://lhchuong.files.wordpress.com/2013/09/untitled3.png>

### Hàm 5

Như vậy là tôi vừa hoàn thành việc mô tả và cài đặt thuật toán **Dijkstra**, trong bài viết tôi không giải thích chi tiết các bước vì đã có rất nhiều trong các diễn đàn lớn...

Chúc các bạn thành công!

Advertisements

[Report this ad](#)[Report this ad](#)

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

## 5 thoughts on “Thuật toán Dijkstra – Tìm đường đi ngắn nhất dựa vào trọng số”

1. [Nguyễn Đạt](#) nói:

Tháng Ba 31, 2015 lúc 9:26 chiều  
rất hữu ích, tôi đã dựa vào đây để làm đồ án, thanks

**Phản hồi**

2. [henry](#) nói:

Tháng Mười 10, 2016 lúc 10:49 sáng  
INT\_MAX khởi tạo ở đâu vậy b?

**Phản hồi**

o [Chuong Le Hoang](#) nói:

Tháng Mười 14, 2016 lúc 7:04 chiều

Nó trong thư viện limits.h nhé bạn 😊

### **Phản hồi**

3. **viên vu** nói:

Tháng Ba 3, 2017 lúc 11:18 chiều

Kết quả thế này đúng ko bạn: 1, 7, 6, 2, 9, 10

thanks bạn

### **Phản hồi**

4. **nhì** nói:

Tháng Mười Một 16, 2017 lúc 9:32 chiều

cout... em nhập vào nó không chạy dc... e sai cfree nên phải printf... sửa lại sao ạ

### **Phản hồi**

[Blog tại WordPress.com.](#)