***Bài toán tìm đường đi ngắn nhất bằng thuật toán Dijkstra***

1. Phát biểu bài toán

Cho một đồ thị có hướng G=(V,E), một hàm trọng số w: E → [0,∞] và một đỉnh nguồn s. Tìm đường đi ngắn nhất từ đỉnh nguồn s đến mỗi đỉnh của đồ thị.

1. Phân tích bài toán

Thuật toán Dijkstra giải quyết bài toán tìm đường đi ngắn nhất từ một đỉnh xuất phát s nào đó tới mọi đỉnh còn lại của đồ thị. Ý tưởng chính của thuật toán là phát triển các đường đi từ s bằng cách thêm dần các đỉnh mới, ta áp dụng thuật toán tham lam.

Các đỉnh của đồ thị được chia làm 2 loại: các đỉnh đã được đi qua U và các đỉnh chưa đi qua N = V \ U. Xuất phát từ đỉnh s, thì có U = {s} và N = V \ {s}. Với mỗi đỉnh i∈V ta gán di là độ dài đường đi ngắn nhất từ s tới i chỉ qua các đỉnh thuộc U. Nếu từ U không đạt được tới i thì di=+∞.

Tiếp theo ta thêm đỉnh chưa được đi qua v∈N = V \ U và U nếu như nó có ước lượng nhỏ nhất dv<di với mọi i∈N = V \ U. Sau đó tính lại các di.

Quá trình đó lặp lại cho đến khi đường đi đã qua hết mọi đỉnh.

1. Phân tích giải thuật.

Dùng ma trận kề để biểu diễn đồ thị C = (cij), trong đó cij= trọng số của cung (i, j), cij=+∞ nếu không có cung (i, j).

Dùng mảng d[] để ghi lại độ dài đường đi ngắn nhất từ đỉnh s tới đỉnh i đang xét. Xuất phát d[s]=0 và d[i]=cij nếu i kề s, d[j]=+∞ nếu j không kề s.

Để khôi phục lại đường đi từ s tới một đỉnh t nào đó, ta dùng mảng Tr[] để ghi nhãn của nút đứng trước trong đường đi, khởi tạo ban đầu cho các phần tử của Tr[] nhận giá trị 0.

Dạng mã giả:

Procedure Dijkstra

U:={tập đỉnh xuất phát s}

Repeat

Tìm vϵ N= V \ U mà d[v] nhỏ nhất

U:=U + {v};

For mỗi i ϵ N=V\U do

If d[v] + c[v, i]<d[i] then

Begin

d[i]:= d[v]+c[v,i];

Tr[i]:=v;

End;

Until U=V;

1. Đánh giá thời gian thực hiện.

Các thủ tục khởi tạo của mảng có thời gian thực hiên là O(n). Vòng lặp repeat-until được thực hiện (n-1) lần. Ở bước thứ i, các vòng lặp bên trong thực hiện (n-i) lần. Vậy thời gian thực hiện thủ tục là O(n2).