GIẢI THUẬT QUAY LUI

Lời giải của bài toán thường biểu diễn bằng một vec tơ gồm n thành phần x=(x1, x2,...,xn) phải thỏa mãn điều kiện nào đó. Để chỉ ra lời giải x, ta phải xây dựng dần các thành phần lời giải xi.

Tại mỗi bước i:

- Đã xây dựng xong các thành phần x1,...,xi-1

- Xây dựng thành phần xi bằng cách lần lượt thử tất cả các khả năng mà xi có thể lựa chọn:

* Nếu chọn một khả năng j nào đó phù hợp cho xi thì xác định xi theo khả năng j. Thường phải có thêm các thao tác ghi nhận trạng thái mới của bài toán để hỗ trợ cho bước quay lui. Nếu i=n thì ta có được một lời giải, ngược lại thì tiến hành bước i+1 để xác định xi+1.
* Nếu không có một khả năng nào chấp nhận được cho xi thì ta lùi lại bước trước để xác định lại thành phần xi-1

Mô hình của phương pháp quay lui có thể viết bằng thủ tục sau, với n là số bước cần phải thực hiện, k là số khả năng mà xi có thể lựa chọn.

Try(i)

For(j=1→k)

If(xi chấp nhận được khả năng j)

{

Xác nhận xi theo khả năng j;

Ghi nhận trạng thái mới;

If(i<n)

Try(i+1);

Else

Ghi nhận nghiệm;

Trả lại trạng thái cũ cho bài toán;

}

BÀI TOÁN TÌM ĐƯỜNG ĐI TRÊN ĐỒ THỊ (DFS)

1. Nêu bài toán tìm đường đi trên đồ thị theo chiều sâu

G = (V, U) là đơn đồ thị (có hướng hoặc vô hướng). V = { 1,., n} là lập các  
đỉnh, u là tập cạnh (cung). Với s, t є V, tìm tất cả các đường đi từ s đến t.

Ý tưởng thuật toán DFS : Tìm kiếm theo chiều sâu.

Thuật toán DFS tiến hành tìm kiếm trong đồ thị theo chiều sâu. Thuật toán thực hiện việc thăm tất cả các đỉnh có thể đạt được cho tới đỉnh t từ đỉnh s cho trước. Đỉnh được thăm càng muộn sẽ càng sớm duyệt xong(cơ chế LIFO- Vào Sau Ra Trước). Nên thuật toán có thể tổ chức bằng một thủ tục đệ quy quay lui.

2. Mô tả chi tiết thuật toán

Input G=(V,U),s,t

Output Tất cả các đường đi từ s đến t (nếu có)

void DFS(int s) {

for (int u = 1; u<= n; u++)

if (chấp nhận được) {

Ghi nhận nó;

if (u!=t )

DFS(u) ;

Else

In đường đi;

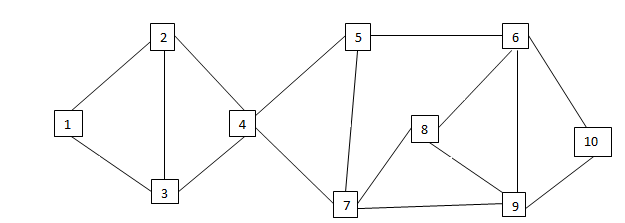
Bỏ việc ghi nhận;

}

}

3. Kiểm tra với 5 bộ dữ liệu

3.1. Bộ dữ liệu 1



Tìm tất cả các đường đi từ đỉnh 1 đến đỉnh 4

Chạy các bước của thuật toán:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thực hiện | Đã xét | Chưa xét | Kết quả |
| 1 | DFS(1) | 1,2 | 3,4,5,6,7,8,9,10 |  |
| 2 | DFS(2) | 1,2,3 | 4,5,6,7,8,9,10 |  |
| 3 | DFS(3) | 1,2,3,4 | 5,6,7,8,9,10 | 1-2-3-4 |
| 4 |  | 1,2,3 | 4,5,6,7,8,9,10 |  |
| 5 |  | 1,2 | 3,4,5,6,7,8,9,10 |  |
| 6 |  | 1,2,4 | 3,5,6,7,8,9,10 | 1-2-4 |
| 7 |  | 1,2 | 3,4,5,6,7,8,9,10 |  |
| 8 |  | 1 | 2,3,4,5,6,7,8,9,10 |  |
| 9 |  | 1,3 | 2,4,5,6,7,8,9,10 |  |
| 10 | DFS(3) | 1,3,2 | 4,5,6,7,8,9,10 |  |
| 11 | DFS(2) | 1,3,2,4 | 5,6,7,8,9,10 | 1-3-2-4 |
| 12 |  | 1,3,2 | 4,5,6,7,8,9,10 |  |
| 13 |  | 1,3 | 2,4,5,6,7,8,9,10 |  |
| 14 |  | 1,3,4 | 2,5,6,7,8,9,10 | 1-3-4 |
| 15 |  | 1,3 | 2,4,5,6,7,8,9,10 |  |
| 16 |  | 1 | 2,3,4,5,6,7,8,9,10 |  |

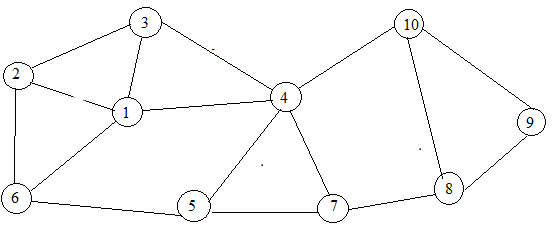
1-2-3-4

1-2-4

1-3-2-4

1-3-4

3.2. Bộ dữ liệu 2



Tìm tất cả các đường đi từ đỉnh 1 đến đỉnh 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thực hiện | Đã xét | Chưa xét | Kết quả |
|  | DFS(1) | 1,2 | 3,4,5,6,7,8,9,10 |  |
|  | DFS(2) | 1,2,3 | 4,5,6,7,8,9,10 |  |
|  | DFS(3) | 1,2,3,4 | 5,6,7,8,9,10 | 1-2-3-4 |
|  |  | 1,2 | 3,4,5,6,7,8,9,10 |  |
|  |  | 1,2,6 | 3,4,5,7,8,9,10 |  |
|  | DFS(6) | 1,2,6,5 | 3,4,7,8,9,10 |  |
|  | DFS(5) | 1,2,6,5,4 | 3,7,8,9,10 | 1-2-6-5-4 |
|  |  | 1,2,6,5 | 3,4,7,8,9,10 |  |
|  |  | 1,2,6,5,7 | 3,4,8,9,10 |  |
|  | DFS(7) | 1,2,6,5,7,4 | 3,8,9,10 | 1-2-6-5-7-4 |
|  |  | 1,2,6,5,7 | 3,4,8,9,10 |  |
|  |  | 1,2,6,5,7,8 | 3,4,9,10 |  |
|  | DFS(8) | 1,2,6,5,7,8,9 | 3,4,10 |  |
|  | DFS(9) | 1,2,6,5,7,8,9,10 | 3,4 |  |
|  | DFS(10) | 1,2,6,5,7,8,9,10,4 | 3 | 1-2-6-5-7-8-9-10-4 |
|  |  | 1,2,6,5,7,8 | 3,4,9,10 |  |
|  |  | 1,2,6,5,7,8,10 | 3,4,9 |  |
|  | DFS(10) | 1,2,6,5,7,8,10,4 | 3,9 | 1-2-6-5-7-8-10-4 |
|  |  | 1,2,6,5,7,8,10 | 3,4,9 |  |
|  |  | 1,2,6,5,7,8,10,9 | 3,4 |  |
|  | DFS(9) | 1 | 2,3,4,5,6,7,8,9,10 |  |
|  |  | 1,3 | 2,4,5,6,7,8,9,10 |  |
|  | DFS(3) | 1,3,2 | 4,5,6,7,8,9,10 |  |
|  | DFS(2) | 1,3,2,6 | 4,5,7,8,9,10 |  |
|  | DFS(6) | 1,3,2,6,5 | 4,7,8,9,10 |  |
|  | DFS(5) | 1,3,2,6,5,4 | 7,8,9,10 | 1-3-2-6-5-4 |
|  |  | 1,3,2,6,5 | 4,7,8,9,10 |  |
|  |  | 1,3,2,6,5,7 | 4,8,9,10 |  |
|  | DFS(7) | 1,3,2,6,5,7,4 | 8,9,10 | 1-3-2-6-5-7-4 |
|  |  | 1,3,2,6,5,7 | 4,8,9,10 |  |
|  |  | 1,3,2,6,5,7,8 | 4,9,10 |  |
|  | DFS(8) | 1,3,2,6,5,7,8,9 | 4,10 |  |
|  | DFS(9) | 1,3,2,6,5,7,8,9,10 | 4 |  |
|  | DFS(10) | 1,3,2,6,5,7,8,9,10,4 | RỖNG | 1-3-2-6-5-7-8-9-10-4 |
|  |  | 1,3,2,6,5,7,8 | 4,9,10 |  |
|  |  | 1,3,2,6,5,7,8,10 | 4,9 |  |
|  | DFS(10) | 1,3,2,6,5,7,8,10,4 | 9 | 1-3-2-6-5-7-8-10-4 |
|  |  | 1,3,2,6,5,7,8,10 | 4,9 |  |
|  |  | 1,3,2,6,5,7,8,10,9 | 4 |  |
|  | DFS(9) | 1,3 | 2,4,5,6,7,8,9,10 |  |
|  |  | 1,3,4 | 2,5,6,7,8,9,10 | 1-3-4 |
|  |  | 1,3 | 2,4,5,6,7,8,9,10 |  |
|  |  | 1 | 2,3,4,5,6,7,8,9,10 |  |
|  |  | 1,4 | 2,3,5,6,7,8,9,10 | 1-4 |
|  |  | 1 | 2,3,4,5,6,7,8,9,10 |  |
|  |  | 1,6 | 2,3,4,5,7,8,9,10 |  |
|  | DFS(6) | 1,6,2 | 3,4,5,7,8,9,10 |  |
|  | DFS(2) | 1,6,2,3 | 4,5,7,8,9,10 |  |
|  | DFS(3) | 1,6,2,3,4 | 5,7,8,9,10 | 1-6-2-3-4 |
|  |  | 1,6 | 2,3,4,5,7,8,9,10 |  |
|  |  | 1,6,5 | 2,3,4,7,8,9,10 |  |
|  | DFS(5) | 1,6,5,4 | 2,3,7,8,9,10 | 1-6-5-4 |
|  |  | 1,6,5 | 2,3,4,7,8,9,10 |  |
|  |  | 1,6,5,7 | 2,3,4,8,9,10 |  |
|  | DFS(7) | 1,6,5,7,4 | 2,3,8,9,10 | 1-6-5-7-4 |
|  |  | 1,6,5,7 | 2,3,4,8,9,10 |  |
|  |  | 1,6,5,7,8 | 2,3,4,9,10 |  |
|  | DFS(8) | 1,6,5,7,8,9 | 2,3,4,10 |  |
|  | DFS(9) | 1,6,5,7,8,9,10 | 2,3,4 |  |
|  | DFS(10) | 1,6,5,7,8,9,10,4 | 2,3 | 1-6-5-7-8-9-10-4 |
|  |  | 1,6,5,7,8 | 2,3,4,9,10 |  |
|  |  | 1,6,5,7,8,10 | 2,3,4,9 |  |
|  | DFS(10) | 1,6,5,7,8,10,4 | 2,3,9 | 1-6-5-7-8-10-4 |
|  |  | 1,6,5,7,8,10 | 2,3,9,10 |  |
|  |  | 1,6,5,7,8,10,9 | 2,3,4 |  |
|  | DFS(9) | 1 | 2,3,4,5,6,7,8,9,10 |  |

Kết quả thu được các đường đi

1-2-3-4

1-2-6-5-4

1-2-6-5-7-4

1-2-6-5-7-8-9-10-4

1-2-6-5-7-8-10-4

1-3-2-6-5-4

1-3-2-6-5-7-4

1-3-2-6-5-7-8-9-10-4

1-3-2-6-5-7-8-10-4

1-3-4

1-4

1-6-2-3-4

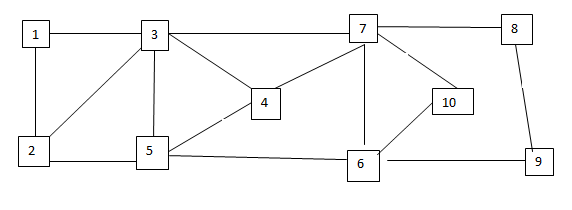
1-6-5-4

1-6-5-7-4

1-6-5-7-8-9-10-4

1-6-5-7-8-10-4

3.3. Bộ dữ liệu 3



4<--3<--2<--1

4<--5<--3<--2<--1

4<--7<--6<--5<--3<--2<--1

4<--7<--8<--9<--6<--5<--3<--2<--1

4<--7<--10<--6<--5<--3<--2<--1

4<--7<--3<--2<--1

4<--5<--6<--7<--3<--2<--1

4<--5<--6<--9<--8<--7<--3<--2<--1

4<--5<--6<--10<--7<--3<--2<--1

4<--3<--5<--2<--1

4<--7<--3<--5<--2<--1

4<--5<--2<--1

4<--3<--7<--6<--5<--2<--1

4<--7<--6<--5<--2<--1

4<--3<--7<--8<--9<--6<--5<--2<--1

4<--7<--8<--9<--6<--5<--2<--1

4<--3<--7<--10<--6<--5<--2<--1

4<--7<--10<--6<--5<--2<--1

4<--5<--2<--3<--1

4<--7<--6<--5<--2<--3<--1

4<--7<--8<--9<--6<--5<--2<--3<--1

4<--7<--10<--6<--5<--2<--3<--1

4<--3<--1

4<--5<--3<--1

4<--7<--6<--5<--3<--1

4<--7<--8<--9<--6<--5<--3<--1

4<--7<--10<--6<--5<--3<--1

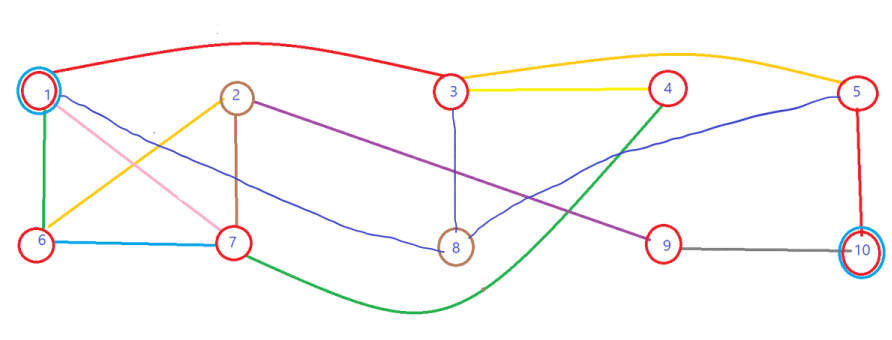
4<--7<--3<--1

4<--5<--6<--7<--3<--1

4<--5<--6<--9<--8<--7<--3<--1

4<--5<--6<--10<--7<--3<--1

3.4. Bộ dữ liệu 4



Tìm tất cả các đường đi từ đỉnh 1 đến đỉnh 4

Kết quả:

4<--3<--1

4<--3<--5<--10<--9<--2<--6<--1

4<--3<--8<--5<--10<--9<--2<--6<--1

4<--3<--5<--10<--9<--2<--7<--6<--1

4<--3<--8<--5<--10<--9<--2<--7<--6<--1

4<--3<--5<--10<--9<--2<--7<--1

4<--3<--8<--5<--10<--9<--2<--7<--1

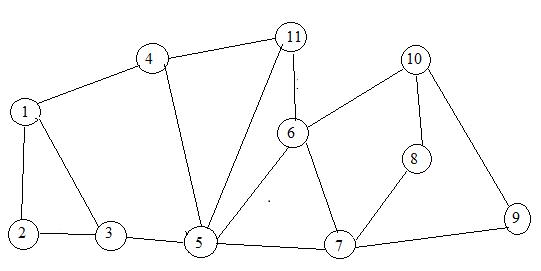
4<--3<--5<--10<--9<--2<--6<--7<--1

4<--3<--8<--5<--10<--9<--2<--6<--7<--1

4<--3<--8<--1

4<--3<--5<--8<--1

3.5. Bộ dữ liệu 5



4<--5<--3<--2<--1

4<--11<--6<--5<--3<--2<--1

4<--11<--6<--7<--5<--3<--2<--1

4<--11<--6<--10<--8<--7<--5<--3<--2<--1

4<--11<--6<--10<--9<--7<--5<--3<--2<--1

4<--11<--5<--3<--2<--1

4<--5<--3<--1

4<--11<--6<--5<--3<--1

4<--11<--6<--7<--5<--3<--1

4<--11<--6<--10<--8<--7<--5<--3<--1

4<--11<--6<--10<--9<--7<--5<--3<--1

4<--11<--5<--3<--1

4<--1

4. Viết chương trình

|  |
| --- |
| #include <conio.h>  #include <stdio.h>  #include<iostream>  using namespace std;  #define MAX 100  #define TRUE 1  #define FALSE 0  /\* tim tat caduong di tu s->t bang phuong phap quay lui.  dung mang truoc[u]=s luu dinh s la dinh truoc u.  neu ko tim dc duong di thi quay lui di duong khac  \*/  int G[MAX][MAX], n, chuaxet[MAX],truoc[MAX];  int t=5,j,s=2;// s la diem bat dau, t la diem ket thuc  //G[i][j]=1 : ton tai duong di tu i->j  void Init(){ // do thi co huong  freopen("DFS.IN", "r", stdin);  cin>>n;  cout<<"So dinh cua ma tran n = "<<n<<endl;  //nhap ma tran lien ke.  for(int i=1; i<=n;i++)  {  for(int j=1; j<=n;j++)  {  cin>>G[i][j];  }  }  }  int xuat()  {  // in 1 duong di tu s->t  cout<<t<<"<--";  j=t;  while(truoc[j]!=s)  {  cout<<truoc[j]<<"<--";  j=truoc[j];  }  cout<<s<<endl;  }  // tim truoc[] chua tat ca cac dih truoc dinh cuoi  // moi lan goi DFS(s,t) tim dc dinh u ke s là gan dinh truoc u=s  void DFS(int G[][MAX], int n, int s, int chuaxet[],int t){  int u;  chuaxet[s]=FALSE; // dinh s da dc xet  for(u=1; u<=n; u++){ // duyet tat ca cac dinh  if(G[s][u]==1 && chuaxet[u]) // neu u chua xet va u noi voi s  {  truoc[u]=s; // truoc u la dinh s  if(u==t) xuat(); // u la dinh cuoi thi in ket qua  else  DFS(G,n,u,chuaxet,t); //truyen u la s di tu u->t  chuaxet[u]=TRUE; // khoi tao tao lai dinh u chua dc xet  }        }  }  int main(void){  Init();  for(int i=1; i<=n; i++)  chuaxet[i]=TRUE; // khoi tao cac dinh deu chua xet  DFS( G,n, s, chuaxet,t); // di tu s-> t  getch();  } |

5. Đánh giá độ phức tạp của thuật toán theo lý thuyết và thực nghiệm