

CSDN 首页 博客 学院 下载 论坛 问答 活动 专题 招聘 APP VIP会员

章法到dinic优化)

12767 已收藏

三塘广路

- a. 如果一条路上的所有边均满足:
- b. 正向边  $f(u,v) < c(u,v)$  —— 反向边:
- c. 则我们称这条路径为一条增广路径。
- d. 好了, 弄懂了一些定义, 接下来就可

既然如此，我们的思路就是：

111

## 一、网络流的定义

有向图  $G=(V,E)$  中:

- 有唯一的一个**源点**S (入度为0: 出发点)
- 有唯一的一个**汇点**T (出度为0: 结束点)
- 图由每条弧  $(u, v)$  和权一非负**容量**  $c(u, v)$

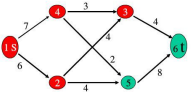
满足 流图。

记为

网络流图是一张只有一个源点和汇点的有向图，而最大流就是求源点到汇点间的最大水流量，下面的问题就是一个最基本，经典的最大流问题

实例：

有一自来水管道输送系统，起点是S，目标是T，途中经过的管道都有一个最大的容量。



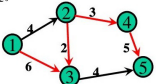
- 问题：问从S到T的最大水流量是多少？

## 二.流量，容量和可行流

对于弧 $(u,v)$ 来说, 流量就是其上流过的水量(我们通常用 $f(u,v)$ 表示), 而容量就是其上可流过的最大水量(我们通常用 $c(u,v)$ 表示), 只要满足 $f(u,v) \leq c(u,v)$ , 我们就称流量 $f(u,v)$ 是可行流(对于最大流问题而言, 所有管道上的流量必须都是可行流)。

### 三.增广路

- 从  $s$  到  $t$  的一条简单路径, 若边  $(u, v)$  的方向与该路径的方向一致, 称  $(u, v)$  为正向边, 方向不一致时称为逆向边。



简单路:  $1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 5$  中。

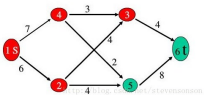
(1, 3) (2, 4) (4, 5) 是正向边, (3, 2) 是逆向边。

如果一条路上的所有边均满足:

正向边:  $f(u,v) < c(u,v)$  —— 反向边:  $f(u,v) > 0$

则我们称这条路径为一条增广路径，简称增广路。

好了，弄懂了一些定义，接下来就可以介绍著名的Ford-Fulkerson算法了。



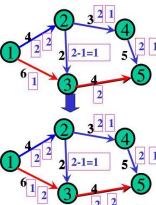
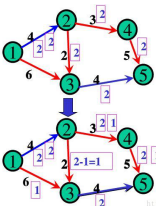
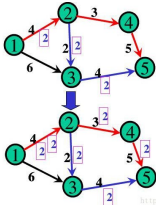
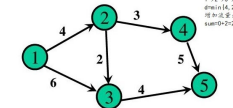
如图所示,如果我们每次都找出一条增广路,只要这条增广路经过汇点,那说明此时水流还可以增加,增加量为 $d(d = \min(d, c(u, v) - f(u, v))$ 或 $d = \min(d, f(u, v))$ ).

我们可以这样理解：对于每一条正向边，他能添加的最大水流为  $c(u,v)-f(u,v)$ 。而对于反向边来说，当正向边上的水流增多时，反向边自身的反向水流会减少，而其能减少的最多水量为  $f(u,v)$ 。由于要保证添加水流之后，所有的  $f(u,v)$  都是可行流，所以我们取最小值。

增加之后，我们要更新流量，每条正向边+ $d$ ，每条反向边- $d$ 即可。

既然这样，我们的思路就是：

1.找出一条增广路径——2.修改其上点的值——3.继续重复1,直至找不出增广路。则此时源点的汇出量即为所求的最大流。

[illegible]

```
44         if(edges[i][k],c<Df)  edges[i][k]=f+wei;
45     else if(edges[i][k],c==Df)  edges[i][k]=f+wei;
46     if(i>=n-1) break;
47     k=i+1;k+=dbs[pre[k+1]];
48     }
49     a[shu[i]-D];
50 }
51 }
52 }
53 void fflow(){
54     int maxFlow=0;
55     for(int i=1;i<=m;i++){
56         for(int j=i;j<=m;j++){
57             if(i==k&&edges[i][j],f<Df)  maxFlow+=edges[i][j],f;
58         }
59     }
60     printf("%d",maxFlow);
61 }
62 }
63 int main(){
64     int n,m,v,f;
65     cin>>n>m>v>f;
66     cin>>n>m>v>f;
67     for(int j=1;j<=m;j++){
68         for(int i=1;i<=m;i++){
69             cin>>edges[i][j],c==Df;
70         }
71     }
72     Ford_Fulkerson();
73     fflow();
74     return 0;
75 }
```

### DINIC算法

Dink算法是网络流最大流的优化算法之一，每一步对原图进行分层，然后用DFS求增广路。时间复杂度是O(n^2\*m)。Dinic算法最多被分为n个阶段，每个阶段包括建层次网络和寻找增广路两部分。

Dink算法的思想是分阶段地在层次网络中增广。它与最短增广路算法不同之处是：最短增广路每个阶段执行完一次BFS增广后，要重新启动BFS从源点s开始寻找另一条增广路;而在Dinic算法中，只需一次BFS过程就可以实现多次增广。

简单来说，分为下面几步：

- 1.在剩余网络中寻找是否存在从S到T的路径，同时构建分层图。
- 2.沿着分层图求增广。
- 3.直到没有可行路径结束算法。

#### 链表版

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int read(){
5     char c;int x;while(c=getchar(),c==' '||c=='\n')continue;
6     while(c=getchar(),c!='0'&&c-'9')x=x*10+c-'0';return x;
7 }
8
9 const int MAXN=10005,MAXM=100005;
10
11 int n,m,s,t,x,y,z,f,t,ans,uze,cnt=0;
12 int u[MAXN*2],v[MAXN*2],w[MAXN*2],q[2*MAXM*2];
13 vector<int> G[MAXN];
14 int to,vail;
15 int head[MAXN];
16
17 void add(int u,int v,int w,int c){
18     G[u].push_back(v);
19     vail++;
20     head[u]=v;
21     w++;
22     u++;
23     G[v].push_back(u);
24     head[v]=u;
25     w++;
26     v++;
27 }
28
29 int BFS(){
30     memset(h,-1,sizeof(h));get(h,s)=1;get(h,t)=1;
31     while(h[s]!=-1){
32         int u=h[s],v=w;uze+=head[v]*w;
33         while(u!=t){
34             if(u!=u1[uzm],u2[uzm][u],v){
35                 if(u!=u1[uzm],u2[uzm][u],v){
36                     q[u]=v;uzm++;
37                 }
38             }
39             uze+=u1[uzm];
40         }
41     }
42     return h[t]>0;
43 }
44
45 int dfs(int now,int N){
46     if(now==t) return N;
47     uze+=now;
48     while(head[now]!=-1){
49         int v=head[now];
50         if(h[v]>h[now],u2[uzm][now]>h[u1[uzm],v]){
51             if(h[v]>h[now],u2[uzm][now]>h[u1[uzm],v]){
52                 uze+=v;
53             }
54             if(h[v]>h[now],u2[uzm][now]>h[u1[uzm],v]){
55                 uze+=v;
56             }
57         }
58         return h[v];
59     }
60     uze+=h[now];
61 }
62 return 0;
63 }
64
65 int main(){
66     int n,m,s,t,x,y,z,f,t,ans,uze,cnt=0;
67     cin>>n>m>s>t>x>y>z>f>t>ans>uze>cnt>0;
68     cin>>n>m>s>t>x>y>z>f>t>ans>uze>cnt>0;
69     for(int i=1;i<=m;i++){
70         int u,v,w;
71         add(x,y,w);
72     }
73     while(BFS()){
74         ans+=dfs(s,t,ans);
75     }
76     printf("%d",ans);
77     return 0;
78 }
```

#### vector版

```
1 #include<iostream>
2 #include<algorithm>
3 #include<vector>
4 #include<string>
5 using namespace std;
6
7 int read(){
8     char c;int x;while(c=getchar(),c==' '||c=='\n')continue;
9     while(c=getchar(),c!='0'&&c-'9')x=x*10+c-'0';return x;
10 }
11
12 const int MAXN=10005,MAXM=100005;
13
14 int n,m,s,t,x,y,z,f,t,ans,uze,cnt=0;
15 int u[MAXN*2],v[MAXN*2],w[MAXN*2],q[2*MAXM*2];
16 int head[MAXN],to,vail,cnt[2*MAXM];
17 vector<int> G[MAXN];
18
19 bool BFS(){
20     int u,v;
21     h[s]=1;
22     while(h[s]!=-1){
23         int u=h[s],v=w;uze+=head[v]*w;
```



网络编程就是构造数据流，每个数据流，其地址是 Unix 域名： <a href="#">#include &lt;netinet/in.h&gt;</a> 。	网络 7	<a href="#">#include &lt;unistd.h&gt;</a>
<b>[网络篇] Link Cut Tree 树分块网络流问题讲解</b>	网络 29	
我在网上看到的一些文章讲解了网络流中的 Dinic 的算法，建议初学者从暴力搜索入手， Dinic 的算法如下	网络 31	
<b>网络流Dinic</b>	网络 31	
<b>最大流 (网络流算法之三,三个算法)</b>	网络 271	
下面是一个简单流式的一系列函数，， 页面地址http://www.program-wiki.com/271/NetworkFlows.html	网络 271	<a href="#">x_3_3_3</a>
<b>[算法] 网络流之最大流--一</b>	网络 2900	
网络流问题， 构造最大流， 网络流是个什么东东呢，今天先写一个最大流的模板记录一下，	网络 2171	<a href="#">MATT1</a>
<b>[网络流入门]最大流算法法</b> 如何求解的	网络 801	
网络流的一些基本术语： 源点，汇点，残差网络，增广路，最大容量约束，网络流值，	网络 815	<a href="#">Pv996</a>
<b>[POJ 1459] 网络流(一) (基本概念和算法)</b>	网络 166	
网络流的一些基本概念和算法(一)可以借助增广路和网络流的一些网络流的一些性质和算法，及其应用，	网络 166	<a href="#">wamr</a>
<b>Dinic 最大流 dinic 算法</b>	网络 166	
Dinic 算法的基本思想：1.根据残差网络构造 G，2.在 G 中寻找增广网络 G 的可行流 F，直到 F 流量达到上界或者 G	网络 166	
typedef long long LType; LType Lf; int Contect; int u = Quere(int u, int v, int flow) { if (u == v) return flow; // 流量达到上界或者 G	网络 240	<a href="#">ethyq</a>
<b>网络流问题的增广与寻找</b>	网络 7687	
为了求解的小问题--最大流，我们从小问题开始--"流量流"。 流量流问题： 从点 u 到点 v 的流量，	网络 7687	<a href="#">Wu Ji</a>
<b>网络流算法总结</b>	网络 263	
<a href="#">网络流算法总结</a>	网络 263	<a href="#">Dream</a>
<b>网络流的一些基本思想 —— Dinic 算法</b>	网络 410	
网络流的一些基本思想：1.根据残差网络构造 G，2.在 G 中寻找增广网络 G 的可行流 F，直到 F 流量达到上界或者 G	网络 410	<a href="#">page271</a>
<b>网络流的增广网络构造化</b>	网络 1616	
1.流量网络构造化：2.网络流网络： E[1:792] Dig and Graphs 可以想成 G，CF866 A，	网络 1616	<a href="#">Fengze</a>
<b>网络流，Dinic 算法</b>	网络 41	
网络流的一些思想： 网络 G 流量流算法的思想： Dinic 算法第一次增广可以求出最大流量，减	网络 41	<a href="#">wamr</a>
<b>网络流： Dinic</b>	网络 171	
网络流的一些基本思想：1.根据残差网络构造 G，2.在 G 中寻找增广网络 G 的可行流 F，直到 F 流量达到上界或者 G	网络 171	<a href="#">Cink1</a>
<b>网络流(最大流问题)</b>	网络 141	
之网络流的一些基本思想：1.根据残差网络构造 G，2.在 G 中寻找增广网络 G 的可行流 F，直到 F 流量达到上界或者 G	网络 141	<a href="#">Cink1</a>
网络流的一些基本思想：1.根据残差网络构造 G，2.在 G 中寻找增广网络 G 的可行流 F，直到 F 流量达到上界或者 G	网络 16	
网络流的一些基本思想：1.根据残差网络构造 G，2.在 G 中寻找增广网络 G 的可行流 F，直到 F 流量达到上界或者 G	网络 418	
网络流的一些基本思想：1.根据残差网络构造 G，2.在 G 中寻找增广网络 G 的可行流 F，直到 F 流量达到上界或者 G	网络 1917	<a href="#">wamr</a>
网络流的一些基本思想：1.根据残差网络构造 G，2.在 G 中寻找增广网络 G 的可行流 F，直到 F 流量达到上界或者 G	网络 1917	<a href="#">wamr</a>
<b>大四学年在学什么，这些基础性的实用工具/学习网站就值得出来了</b>	网络 1729	
在大学中，你肯定需要一些"基础性的实用工具"，对于大学，特别是理工科，肯定需要用到一些"基础性的实用工具"，	网络 1729	
<b>在代码中你是否有过疑问?</b>	网络 1102	
网络，你肯定需要一些"基础性的实用工具"，对于大学，特别是理工科，肯定需要用到一些"基础性的实用工具"，	网络 1102	
网络，你肯定需要一些"基础性的实用工具"，对于大学，特别是理工科，肯定需要用到一些"基础性的实用工具"，	网络 1102	
Java C++ Python C# CF Visual Basic .NET JavaScript PHP SCL Golang 网络 Assembly language Swift Ruby MATLAB Julia C++ Fort Visual Basic Objective-C Delphi Objective-C++	网络 1102	