

permui  
Funny.

首页 新随笔 管理 随笔 - 143 文章 - 0 评论 - 7

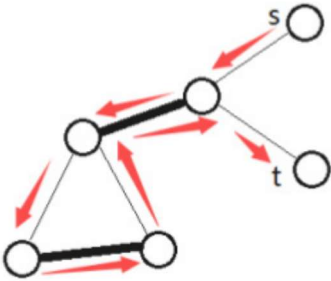
一般图最大匹配-带花树算法

问题

对于一个图 $G(V, E)$ ，它的匹配 $M$ 是二元组 $(u, v)$ 组成的集合，其中 $u, v \in V, (u, v) \in E$ ，并且 $M$ 中不存在重复的点。

当 $|M|$ 最大的时候，我们称 $M$ 为 $G$ 的最大匹配。

当 $G$ 是一个二分图的时候，它的最大匹配可以用经典的匈牙利算法或网络流算法求解。然而当 $G$ 是一个一般的图时，直接进行增广就变得不可行了，例如下面这个例子（论文中的图）：



这个问题出现的原因，就是一个一般图中会含有奇环，即一个点数为 $2k + 1, k > 0$ 的环，而如果经过一个奇环，那么会得到两条含有同一个点的匹配边，这其实是不符合定义的。那为什么二分图可以直接增广呢？因为二分图中不可能含有奇环，它所有的环都是偶环。因此，在一般图匹配问题中，我们需要一种改进算法来解决奇环的问题。

算法

基本算法依然是分为 $n$ 个阶段寻找增广路。问题主要在奇环上，那么我们分析一下这个奇环的性质。首先，奇环中有 $2k + 1$ 个点，所以最多有 $k$ 组匹配。这就是说，有一个点没有匹配，即这个点在环内两边的连边都不是匹配边，也只有这个点可以向环外连边。

发现了这个性质，我们可以把整个奇环缩成一个点。缩完点后的图如果可以找到一条增广路，那么原图中也可以找到一条增广路，因为如果增广路经过奇环那么奇环内的增广路可以还原出来。

这就是带花树算法的思想。整个求解过程分为 $n$ 个阶段，每个阶段从没有匹配的 $s$ 点开始bfs找增广路。搜索的开始，把 $s$ 点加入队列中，标记它为 $A$ 类点。如果从 $x$ 点出发，搜索到了一个未标记的点，有两种情况。如果这个未标记点有匹配，那么把这个点设为 $B$ 类点，它的匹配点设为 $A$ 类点，加入队列继续增广。如果这个点没有匹配，又因为我们是从一个未匹配点开始进行搜索的，所以这说明我们找到了一条增广路，沿着过来的边找回去，展开带花树，修改搜索的过程中，如果我们遇到了偶环，那么不管它，因为它不会影响求解。如果遇到了一个奇环，那么我们找到当前点 $x$ 和找到的点 $v$ ，求出他们的最近公共花祖先，然后把环缩掉。这里我们用并查集实现。

我们在缩环的时候，处理出一个 $pre$ 数组，表示我们回跳的时候走到这里该往哪一个方向走回去。回跳的时候，每次找到 $pre$ ，然后修改这条边，接着跳到 $pre$ 原来的 $match$ 处。如果我们倒着进入一个花的时候，上方的边为非匹配边，那么我们会往下走，这个时候 $pre$ 就应该往下设。中间相遇的位置 $pre$ 互相连接，

```
pre[x]=y,pre[y]=x。
```

算法分为 $n$ 个阶段，每个阶段最多把整个图遍历一次，每个点会最多被缩 $n$ 次花，所以总复杂度为 $O(n^3)$ 。

代码

uoj79，一般图匹配模板题。

```
#include<cstdio>
#include<cctype>
#include<cstring>
```

搜索

我的标签

- 动态规划(26)
- 数学(14)
- 线段树(11)
- 字符串(11)
- 树(10)
- 概率与期望(8)
- 后缀自动机(8)
- 连通性状压dp(8)
- 高斯消元(7)
- 网络流(7)
- 更多

随笔档案

- 2018年4月 (1)
- 2018年1月 (1)
- 2017年11月 (2)
- 2017年10月 (7)
- 2017年9月 (16)
- 2017年8月 (19)
- 2017年7月 (25)
- 2017年6月 (1)
- 2017年5月 (13)
- 2017年4月 (58)

Teammates

- GJS
- HGR
- HJH
- LYY
- Sdchr
- sshockwave
- SSR
- XJ
- ZWL

最新评论

- 1. Re:可持久化Treap  
请问一下，为什么split操作中右子树merge后 now要pushdown呢  
--Adscn
- 2. Re:bzoj3625-小朋友和二叉树  
不过转移k那里有点问题  
--nosta

```

#include<algorithm>
using namespace std;
int read() {
    int x=0,f=1;
    char c=getchar();
    for (;!isdigit(c);c=getchar()) if (c=='-') f=-1;
    for (;isdigit(c);c=getchar()) x=x*10+c-'0';
    return x*f;
}
const int maxn=505;
const int maxm=maxn*maxn*2;
int n,m,que[maxm],ql,qx,pre[maxn],tim=0;
struct edge {
    int v,nxt;
} e[maxm];
int h[maxn],tot=0;
int match[maxn],f[maxn],tp[maxn],tic[maxn];
int find(int x) {
    return f[x]==x?f[x]:f[x]=find(f[x]);
}
void add(int u,int v) {
    e[++tot]=(edge){v,h[u]};
    h[u]=tot;
}
int lca(int x,int y) {
    for (++tim;swap(x,y)) if (x) {
        x=find(x);
        if (tic[x]==tim) return x; else tic[x]=tim,x=pre[match[x]];
    }
}
void shrink(int x,int y,int p) {
    while (find(x)!=p) {
        pre[x]=y,y=match[x];
        if (tp[y]==2) tp[y]=1,que[++qx]=y;
        if (find(x)==x) f[x]=p;
        if (find(y)==y) f[y]=p;
        x=pre[y];
    }
}
bool aug(int s) {
    for (int i=1;i<=n;++i) f[i]=i;
    memset(tp,0,sizeof tp),memset(pre,0,sizeof pre);
    tp[que[ql=qx=1]=s]=1; // 1: type A ; 2: type B
    int t=0;
    while (ql<=qx) {
        int x=que[ql++];
        for (int i=h[x],v=e[i].v;i;i=e[i].nxt,v=e[i].v) {
            if (find(v)==find(x) || tp[v]==2) continue;
            if (!tp[v]) {
                tp[v]=2,pre[v]=x;
                if (!match[v]) {
                    for (int now=v,last,tmp;now;now=last) {
                        last=match[tmp=pre[now]];
                        match[now]=tmp,match[tmp]=now;
                    }
                    return true;
                }
                tp[match[v]]=1,que[++qx]=match[v];
            } else if (tp[v]==1) {
                int l=lca(x,v);
                shrink(x,v,l);
                shrink(v,x,l);
            }
        }
    }
    return false;
}
int main() {
#ifdef ONLINE_JUDGE
    freopen("test.in","r",stdin);
    freopen("my.out","w",stdout);
#endif
    n=read(),m=read();
    for (int i=1;i<=m;++i) {
        int x=read(),y=read();
        add(x,y),add(y,x);
    }
    int ans=0;
    for (int i=1;i<=n;++i) ans+=(!match[i] && aug(i));
    printf("%d\n",ans);
    for (int i=1;i<=n;++i) printf("%d ",match[i]);
}

```

3. Re:bzoj3625-小朋友和二叉树  
受益匪浅!

--nosta

4. Re:斯特林数  
@3C\_KMnO4已修正, 谢谢...

--permui

5. Re:斯特林数  
第二类斯特林数的第4条性质有误,应该是  
 $2^{n-1}-1$ .如 $s(2,2)=1$ ;  
--3C\_KMnO4

## 阅读排行榜

1. 最大流算法-ISAP(4508)
2. 一般图最大匹配-带花树算法(4251)
3. 最大流算法-最高标号预流推进(HLPP)(4015)
4. 斯特林数(2123)
5. Lucas定理(1803)

## 评论排行榜

1. bzoj3625-小朋友和二叉树(2)
2. 斯特林数(2)
3. 最大流算法-最高标号预流推进(HLPP)(1)
4. 最大流算法-ISAP(1)
5. 可持久化Treap(1)

```
puts("");
return 0;
}
```

标签： 带花树

好文要顶

关注我

收藏该文

permui

关注 - 0

粉丝 - 8

2

0

+加关注

« 上一篇: [最大流算法-最高标号预流推进\(HLPP\)](#)  
» 下一篇: [bzoj2429- 聪明的猴子](#)

posted @ 2017-05-15 21:19 permui 阅读(4251) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码
- 【前端】SpreadJS表格控件，可嵌入系统开发的在线Excel
- 【推荐】码云企业版，高效的企业级软件协作开发管理平台
- 【推荐】程序员问答平台，解决您开发中遇到的技术难题



相关博文:

- [【learning】一般图最大匹配——带花树](#)
- [一般图最大匹配-带花树算法](#)
- [【Learning】带花树——一般图最大匹配](#)
- [一般图最大匹配——带花树](#)
- [一般图最大匹配——带花树算法学习小记](#)



最新新闻:

- [他假装病人住进精神病院，揭开了精神病学诊断的黑历史](#)
  - [谷歌进军电信？拟联手Dish成立美国第四大运营商](#)
  - [新思科技助力，三星5nm/4nm/3nm工艺再加速](#)
  - [华为在泰国手机市场份额增长至20%，鸿蒙有没有具体时间表](#)
  - [联发科首发旗舰智能电视芯片S900：支持8K视频解码](#)
- » [更多新闻...](#)