

# BZOJ 3143 HNOI2013 游走 期望DP+高斯消元

标签: BZOJ BZOJ3143 期望DP 高斯消元

2014-12-29 09:44

899人阅读

评论(0)

收藏

举报

分类: 期望DP (13) 高斯消元 (21) BZOJ (651)

版权声明：本文为博主原创文章，未经博主允许不得转载。

题目大意：给定一个无向连通图，我们需要给每条边附一个 $1\sim m$ 的不重复的权值，使 $1$ 到 $n$ 的期望权值和最小  
首先贪心思想是求出每条边的期望经过次数 然后对期望值最小的边附加 $m$ 的权值，第二小的边附加 $m-1$ 的权值，以此类推。

令 $f[i]$ 为第 $i$ 个点的期望经过次数 那么每条边的期望经过次数就是 $f[x]/d[x]+f[y]/d[y]$  其中 $d[x]$ 表示 $x$ 的度数  
那么显然有：

$$f[1]=1+\sum_{j=1}^{n-1} f[j]/d[j]$$

$$f[i]=\sum_{j=i+1}^n f[j]/d[j] \quad (2\leq i\leq n-1)$$

其中 $f[n]$ 应该等于 $1$ ，但是由于进入了点 $n$ 之后就不再出来，因此 $n$ 点对周围的边和点的贡献都是 $0$ ，不妨令 $f[n]=0$

那么就有 $n$ 个未知数 $n$ 个方程 高斯消元求解即可

然后代入即可出解 注意求的是最小值不是最大值 因此最小的期望值应该乘最大的边权 不要弄反  
边集又开小了 - - 我是怎么了最近 - -

[cpp]

```
01. #include <cmath>
02. #include <cstdio>
03. #include <cstring>
04. #include <iomanip>
05. #include <iostream>
06. #include <algorithm>
07. #define M 550
08. using namespace std;
09. struct abcd{
10.     int to,next;
11. }table[M*M<<1];
12. int head[M],tot;
13. int n,m,degree[M];
14. double f[M][M],a[M];
15. double expectation[M*M],ans;
16. void Add(int x,int y)
17. {
18.     degree[x]++;
19.     table[++tot].to=y;
20.     table[tot].next=head[x];
21.     head[x]=tot;
22. }
23. void Gauss_Elimination()
```

```
24. {
25.     int i,j,k;
26.     for(i=1;i<=n;i++)
27.     {
28.         k=i;
29.         for(j=i+1;j<=n;j++)
30.             if( fabs(f[j][i])>fabs(f[k][i]) )
31.                 k=j;
32.         for(j=i;j<=n+1;j++)
33.             swap(f[i][j],f[k][j]);
34.         for(k=i+1;k<=n;k++)
35.         {
36.             double temp=-f[k][i]/f[i][i];
37.             for(j=i;j<=n+1;j++)
38.                 f[k][j]+=f[i][j]*temp;
39.         }
40.     }
41.     for(i=n;i;i--)
42.     {
43.         for(j=i+1;j<=n;j++)
44.             f[i][n+1]-=f[i][j]*a[j];
45.         a[i]=f[i][n+1]/f[i][i];
46.     }
47. }
48. int main()
49. {
50.     int i,x,y;
51.     cin>>n>>m;
52.     for(i=1;i<=m;i++)
53.     {
54.         scanf("%d%d",&x,&y);
55.         Add(x,y),Add(y,x);
56.     }
57.     for(x=1;x<n;x++)
58.     {
59.         f[x][x]-=1;
60.         if(x==1) f[x][n+1]=-1;
61.         for(i=head[x];i;i=table[i].next)
62.             f[x][table[i].to]+=1.0/degree[table[i].to];
63.     }
64.     f[n][n]=1;f[n][n+1]=0;
65.     Gauss_Elimination();
66.     for(i=1;i<=m;i++)
67.     {
68.         x=table[i+1-1].to;
69.         y=table[i<1].to;
70.         expectation[i]=a[x]/degree[x]+a[y]/degree[y];
71.     }
72.     sort(expectation+1,expectation+m+1);
73.     for(i=1;i<=m;i++)
74.         ans+=(m-i+1)*expectation[i];
75.     cout<<fixed<<setprecision(3)<<ans<<endl;
76.     return 0;
77. }
```

