HDU 3018 Ant Trip(欧拉回路:一笔画问题)

<http://acm.hdu.edu.cn/showproblem.php?pid=3018>

http://blog.csdn.net/u013480600/article/details/30285541

**题意:**

        给你无向图的N个点和M条边,保证这M条边都不同且不会存在同一点的自环边,现在问你至少要几笔才能所有边都画一遍.(一笔画的时候笔不离开纸)

**分析:**

        首先根据给出的边我们只需要分别处理每个连通分量需要多少笔即可.

        如果该连通分量是一个孤立的点,显然只需要0笔.

        如果该连通分量是一个欧拉图或半欧拉图,只需要1笔.

        现在关键是连通分量并非一个(半)欧拉图时,需要几笔?

**一般性的结论是:**

**非(半)欧拉图需要的笔数==该图中奇数度的点数目/2**

       下面来证明该结论:

       首先一个无向图的连通分量中的奇数度的点个数一定是偶数个(成对出现),因为无向图的总度数=偶数.

       我们在这种连通分量中每次画一笔有两种选择:

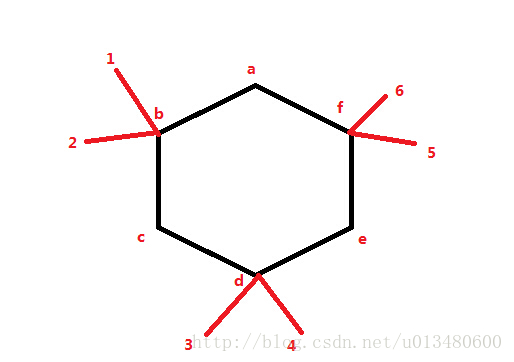
       1.a->b->c->d…->g    一条起点与终点不同的路径(**路中除首尾度减1外,每个点度减2)**

       2.a->b->c->d…->a    一条起点与终点相同的回路**(路中每个点度数减2)**

       也就是说想要把**非(半)欧拉图分量**中的奇数度的点的度数都变成偶数,我们**至少需要画**奇数度点个数/2 笔.

       那么对于这种图我们**最多需要画**的笔数 是不是也是 : 奇数度点个数/2 笔呢?

       答案是肯定的,这里假设图中有4个奇数度的点,1,2,3,4,5,6.如下图所示:



       我们先走路:1-> b-> c-> d-> e-> f->a-> 2 这条路.然后6,5 和3,4 分别属于两个连通分量了.可以看出只需要3笔,即6/2=3即可.

       当然也可以这么画:1->b->2 然后6->f->5 ,接着就剩下了一个半欧拉图了,半欧拉图也是1笔画,正好3笔.

**也就是说对于这种非(半)欧拉图的连通分量,我们每笔必然消除正好2个点的奇度(使其度变偶数),当最后一笔的时候我们必然消除所有的点的度数(包括剩下的两个奇点,因为最后一笔就必然是欧拉通路).但是有一点要注意,有可能过程中的某几笔会使得该连通分量变成多个连通分量,当然结论不变.**

       经过上面的分析,这题的结论出来了,对于每个以i为根的连通分量我们记录属于该连通分量的点数目num[i]和该连通分量中奇度点的个数odd[i].

       如果num[i]==0或1,需0笔.(注意num[i]==0表示i点不是根,num[i]==1表示i点是一个孤立的点.)

       如果num[i]>1且odd[i]==0 需1笔

       如果num[i]>1且odd[i]>0 需odd[i]/2笔

**AC代码:**

**[cpp]** [view plain](http://blog.csdn.net/u013480600/article/details/30285541) [copy](http://blog.csdn.net/u013480600/article/details/30285541)

[print?](http://blog.csdn.net/u013480600/article/details/30285541)

1. #include<cstdio>
2. #include<cstring>
3. using namespace std;
4. const **int** maxn=100000+10;
5. **int** n,m;
6. **int** fa[maxn];
7. **int** num[maxn],odd[maxn],degree[maxn];
8. **int** findset(**int** i)
9. {
10. if(fa[i]==-1) return i;
11. return fa[i]=findset(fa[i]);
12. }
13. **int** main()
14. {
15. while(scanf("%d%d",&n,&m)==2)
16. {
17. memset(fa,-1,sizeof(fa));
18. memset(num,0,sizeof(num));
19. memset(odd,0,sizeof(odd));
20. memset(degree,0,sizeof(degree));
21. for(**int** i=0;i<m;i++)
22. {
23. **int** u,v;
24. scanf("%d%d",&v,&u);
25. degree[v]++;
26. degree[u]++;
27. u=findset(u), v=findset(v);
28. if(u!=v) fa[u]=v;
29. }
30. for(**int** i=1;i<=n;i++)
31. {
32. num[findset(i)]++;      //num[i]=x表以i为根的连通分量中有x个节点
33. if(degree[i]%2) odd[findset(i)]++;//odd[i]=x表以i为根的连通分量中有x个奇度的点
34. }
35. **int** ans=0;
36. for(**int** i=1;i<=n;i++)
37. {
38. if(num[i]<=1) continue;
39. else if(odd[i]==0) ans++;
40. else if(odd[i]>0) ans+=odd[i]/2;
41. }
42. printf("%d\n",ans);
43. }
44. return 0;
45. }