2017省选模拟赛之网络流专题

解题报告

长沙市长郡中学 刘俊琦

Problem 1 . 奶牛议会

这道题与“和平委员会”较为类似，可以使用2-SAT解决。

对于每一份投票，如果不满足其中一个，就一定要满足另一个。可以将题目中给出的议案拆分成通过和不通过两个点，用有向图表示它们之间的依赖关系。例如投票1Y2N，就从1N向2N连边，从2Y向1N连边。我们在做决定的时候每选择一个点，就必须选择它的所有后继节点。

从1号议案开始，对于每一个没有确定的议案，首先以1Y为根节点进行dfs，将所有访问到的点标记为选择并压入栈中。如果在过程中出现冲突，也就是一个点被访问的时候，它的对应点已经被标记为选择了，那么就利用栈中存储的点信息恢复之前的修改，改从1N为根节点进行dfs，同上操作。如果依旧出现冲突，则说明问题无解，否则这个议案就是N。如果第一次dfs的过程中没有出现冲突，那么先恢复修改，但不清空记录点的栈，以1N为根节点再进行一次dfs，若冲突，则这个议案为Y，否则这个议案为？，并恢复修改。也就是说，标记为问号的议案不修改任何节点的状态，因为它不会对答案造成影响。可以证明，这种算法能够保证答案的正确性。

Problem 2 . 基因串

这道题虽然出现在网络流专题，但是和网络流并没有关系。

对于这一类型的问题，区间DP往往是一个较好的选择。定义g[i][j]为第i位至第j位最少用几个S可以生长到。根据经验，转移是枚举断点。即：

g[i][j]=min{g[i][d]+g[d+1][j]} (i<=d<j)

我们需要考虑边界情况的处理。什么情况下答案是确定的呢？如果i位到j位组成的子串可以由一个S生长得到，那么答案就是1。如果区间长度为1，若这一位就是S，那么答案就是1，否则答案是0。现在问题便转化成区间i到j是否能够由1个S生长得到。定义f[i][j]为i到j是否可以由S生长得到，如果由一个S生长得到，那么为1，如果只能由多个S生长得到，那么为2（方便判断NIE的情况）。

继续考虑枚举断点，需要将一个区间的两个部分合并为总的答案。可以发现，如果左右两部分都可以由一个S生长得到，那么答案便可能为2。什么情况下答案为1呢？如果S可以生长成X和Y，那么当左半区间可以由一个X生长得到，右半区间可以由一个Y生长得到，答案就是1了。经过这样分析，我们意识到状态的设计有一些问题。不妨再加上一维c，定义f[i][j][c]为i到j是否可以由c字符生长得到，如果由一个c生长得到，那么为1，如果只能由多个c生长得到，那么为2。转移过程不再赘述，注意要取到最优，也就是既可能为2有可能为1的时候答案应为1。记忆化搜索是实现的不错选择。

在主函数中，我们先判断f[1][len][c]是否为零，以判断NIE，然后输出g[1][len]的值作为最终的答案。

Problem 3 . 最优标号

这是胡伯涛《最小割模型在信息学竞赛中的应用》中的一道例题。

注意到xor运算的位无关性，可以把编号按二进制位处理，答案根据二进制位累加。下面的讨论就认为点的编号只有0和1两种选择。

注意到选择是两种，可以联想到最小割模型。如果一个点为0，就令它在S割中，如果一个点为1，就令它在T割中。这样，所有两个点分居S和T两割的边都对答案有1的贡献。如果把所有边的两端的点连上双向容量为1的边，那么最终最小割就是答案。由于题目中给出了一部分点的权值，所以要从S向所有为0的点连inf的边，从所有为1的点向T连inf的边，这样可以强制这些点在S割或者T割。

进行上述转化后，可以使用Dinic算法求解最终答案。

Problem 4 . 美食节

看到这道题，可以想到四川2007年的省选题“修车”，但本题的数据范围更大。

20%算法

先不考虑一道菜被点了多次，把每次点菜都单独来处理，套用修车的思路，将厨师拆点并连边。可以通过1号和4号测试点。

60%算法

注意到一道菜被点了多次的时候有很多重复的点，可以把重复的点合在一起，一道菜作为一个点，从源点连容量为被点总数的边。这样可以显著提升运行效率。

100%算法1

把费用流从SPFA改成Dijkstra后即可通过全部测试点。

100%算法2

一开始不把所有边连上，只连接每个厨师的第一道菜与每道菜的边和第一道菜与T的边。每次对于当前增广的厨师连接它的下一道菜与每道菜和T的边。这样可以让SPFA运行得更快，虽然没有复杂度上的区别，但实际结果非常好。