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：

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

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。