PKUWC2018解题报告

Day1

Minimax

考虑动态规划，先将叶子的权值离散化，假设权值现在是$1..m$

令$f[x][i]$表示结点x的权值是i的概率。

若x是叶子，则令$f[x][val[x]]=1$即可。

若x只有一个儿子，设它是y，则令$f[x][i]=f[y][i]$。

若x有两个儿子y,z，则考虑用线段树记录$f[y][i],f[z][i]$，然后用线段树合并来计算答案。

考虑线段树合并，假设现在合并$f[y][l..r],f[z][l..r]$，我们可以沿途记录下：

$$\sum_{i=1}^{l-1}f[z][i],\sum_{i=1}^{l-1}f[y][i],\sum_{i=r+1}^{m}f[z][i],\sum_{i=r+1}^{m}f[y][i]$$

当某一方的结点为空时，可以打下一个乘法标记来更新。

时间复杂度：$O(n\log m)$

Slay the Spire

可以发现，由于强化牌的数字都大于1，那我们肯定是尽量使用强化牌，然后最后打出最大的攻击牌，所以可以分类讨论以下几个情况：

假设选了A张强化牌，B张攻击牌（A+B=m）。

1.如果$A>K-1$，那么肯定是选最大的$K-1$张强化牌，然后选最大的攻击牌。

2.如果$A<K-1$，那么肯定是选所有的强化牌，然后选最大的$K-A$张攻击牌。

对于$K=1$要独立讨论。

对于第一种情况，可以用dp算出$f[i]$：选了i张强化牌，最大的$K-1$张的乘积的和。然后枚举最大的攻击牌是啥用排列组合计算方案。

对于第二种情况，可以发现我们选的攻击牌是最大的$B-(M-K)$张，我们可以令$g[i]$表示攻击牌选了i张，最大的$i-(M-K)$张的和的期望，这个也可以在$O(n^2)$内算好。

时间复杂度：$O(n^2)$

斗地主

考虑先把所有能春天的手牌预处理出来，分类讨论：

1.剩下的牌可以组成炸弹

2.剩下的牌不能组成炸弹

对于第一种情况：

地主的手牌组成肯定是：若干副炸弹+一个可以一次打完的牌，且这些炸弹要比剩下的牌组成的炸弹大。

这个大力爆搜就行了

对于第二种情况：

首先因为剩下的牌不能组成炸弹，所以所有牌都至少有一张，所以有14张确定了。

剩下6张继续大力搜。

一共只有30000多种答案。

Day2

随机算法

令$f[S]$表示，现在S集合里的点已经无法被加到独立集中了。

枚举一个点$x(x \notin S)$，然后更新$S = S \cup \{x\} \cup adj[x]$。（因为无法被加入独立集中的点我们已经无所谓他在排列的哪边了）。

转移的权值是$\frac{1}{n-|S|}$（也就是剩下$not\ S$中，x在他们中的第一个的概率）。

至于要保证是最大独立集。可以预处理出所有点集的最大独立集，转移时如果从$f[S]$转移到$f[T]$，检查一下$max[not\ T] + 1 = max[not\ S]$即可。

时间复杂度：$O(n2^n)$

猎人杀

相当于求对于所有满足$p[1] = 1$的排列，求：

$$\prod_{i=1}^{n} \frac{w[p[i]]}{\sum_{j=1}^{i} w[p[j]]}$$

并求和。

假设没有$p[1] = 1$的限制的话，那么和就是1

也就是说：

$$\sum_{p} \prod_{i=1}^{n} \frac{1}{\sum_{j=1}^{i} w[p[j]]} = \prod_{i=1}^{n} \frac{1}{w[i]}$$

我们理解一下其中的组合意义：

有一堆球，第i种颜色的球有$w[i]$个，球两两之间是不同的，每种颜色的球里有一颗比较特殊的叫，我们称之为大球。

上面这个式子相当于：将所有球排成一列，要求每个大球必须都在和它颜色相同的球的前面。

右边的式子现在算的是这个的概率，前面的式子相当于枚举每种颜色第一颗球的出现顺序($p[n]$表示第一个出现的颜色，以此类推)，然后去计算。

于是根据这个，我们就知道$p[1]=1$意味着，颜色为1的球要在最后出现。

我们枚举颜色为1的大球的位置为d，那么后面还有$w[1]-1$个颜色为1的球，剩下$m-d-(w[1]-1)$个位置要放其他颜色的球，且其他颜色的球不能全放进来，否则颜色1的球就不是最后出现的了。

于是我们可以写出第i种颜色的球的生成函数：

$$f_i(x)=\sum_{j=0}^{w[i]-1}\frac{x^j}{(w[i]-j)*j!}$$

然后把他们乘起来就行了，用分治fft即可。

时间复杂度；$O((\sum w)*\log n*\log\sum w)$

随机游走

令$c[i]$为第i个点在哪个时刻被经过，相当于求：

$$E[Max_{x\in S}c[x]]$$

令$max(S)=Max_{x\in S}c[x]$

令$min(S)=Min_{x\in S}c[x]$

则有经典的$min-max$容斥：

$$E[max(S)]=\sum_{T\subseteq S}(-1)^{|T|-1}E[min(T)]$$

考虑预处理出所有$E[min(T)]*(-1)^{|T|-1}$，然后用子集卷积可以算出所有$E[max(S)]$

考虑计算$E[min(S)]$，相当于只要到达了S中的某个点的就停止。

朴素的暴力是高斯消元，时间复杂度是$O(n^3)$的

但是这里我们可以用树形dp，以起点作为根，对于非根结点x计算从x出发在x的子树里停止的概率和回到$fa[x]$的概率。

这个可以通过列方程在$O(n)$时间内算出来。

所以时间复杂度：$O(n2^n+Qn)$