OpenTrain及Topcoder题目选讲

金胜蓦

NEERC 2015 J. Jump（交互题）

- 有一个长度为n的01串S，每次可以询问一个长度为n的01串T。如果T和S有n个位置相同则返回n，如果T和S有n/2个位置相同则返回n/2，其他情况返回0。在n+500步内确定S。
- n <= 1000，n是偶数。

Solution

- $\frac{\binom{n}{n/2}}{2^n} > 0.01$，随机生成字符串$\mathrm{T}_0$，大约做100次就能找到一个和S匹配位数恰好为$\frac{n}{2}$的$\mathrm{T}_0$。

Solution

- $\frac{\binom{n}{n/2}}{2^n} > 0.01$，随机生成字符串$\mathrm{T}_0$，大约做100次就能找到一个和S匹配位数恰好为$\frac{n}{2}$的$\mathrm{T}_0$。

- 现在T_0中有两种位置，一种是正确的位置，一种是错误的位置，每种位置都恰有$\frac{n}{2}$个。如果把位置分成两个集合，那么同时修改其中一个集合中的位置，尝试两次就能得到S。

Solution

- 如果同时修改T_0的第i个位置和第j个位置，得到的结果不是$\frac{n}{2}$，说明位置i和位置j同时正确或者同时错误。

Solution

- 如果同时修改T_0的第i个位置和第j个位置，得到的结果不是$\frac{n}{2}$，说明位置i和位置j同时正确或者同时错误。
- 同时位置1和位置i的字符，可以知道i和1是不是属于同一类的位置，询问n-1次就能得到两个集合。

Solution

- 如果同时修改T_0的第i个位置和第j个位置，得到的结果不是$\frac{n}{2}$，说明位置i和位置j同时正确或者同时错误。
- 同时位置1和位置i的字符，可以知道i和1是不是属于同一类的位置，询问n-1次就能得到两个集合。
- 期望次数：n + 100

XVI Open Cup named after E.V. Pankratiev. GP of Ekaterinburg J. Jarvis

- 给定一张n个点的带权无向完全图，以及m个限制(s, t, l)，代表从s到t的所有路径长度都为l（可以走重复的边，边权可为负）。问，如何给图中的边定权值，使得满足所有限制。

- n, m <= 150

Solution

- 首先，所有的环长度必须为0。如果有个环权值不为0，那就可以通过走那个环使得路径长度不唯一。

Solution

- 首先，所有的环长度必须为0。如果有个环权值不为0，那就可以通过走那个环使得路径长度不唯一。
- 给每个节点打个顶标Vi，令边的权值w(x,y) = Vy – Vx，这样从s到t的路径长度唯一等于Vt - Vs，并且能够保证每个环的长度恒为0。

Solution

- 原题中的限制(s,t,l)就相当于：Vt - Vs = l。稍微转换一下就变成了：Vs + l >= Vt, Vt - l <= Vs。然后就是差分约束问题了。

NEERC 2014 E. Epic Win!

- A和B两个人玩石头剪刀布的游戏，现在已知B的策略可以用一个自动机表示。自动机的每个节点为(C, R, P, S)，C代表本轮游戏中出剪刀、石头还是布，R、P、S代表本轮游戏中对方出剪刀、石头、布之后，转移到的状态。唯一不能确定的是该自动机的开始节点。现在要求构造一个A的自动机，使得玩了10亿轮之后，A的胜率可以达到99%。
- B的自动机节点个数不超过100，A的自动机节点个数不超过50000。

Solution

- 为A的自动机的每个节点加一个属性，代表当A到达该节点时，B可能到达节点集合S。

Solution

- 为A的自动机的每个节点加一个属性，代表当A到达该节点时，B可能到达节点集合S。
- 根据S中节点的策略，如果是唯一的，则A出一个能够打败B的动作。

Solution

- 为A的自动机的每个节点加一个属性，代表当A到达该节点时，B可能到达节点集合S。
- 根据S中节点的策略，如果是唯一的，则A出一个能够打败B的动作。
- 如果不是唯一的，则A随便出一个动作，再根据对手的动作将当前的集合分裂，得到若干个新的集合。

Solution

- 不断地迭代，直到A的自动机节点不再更新为止。可以证明每个节点要么在n步之内发生分裂，要么永远不会发生分裂。（n是B的自动机节点个数）。

Solution

- 不断地迭代，直到A的自动机节点不再更新为止。可以证明每个节点要么在n步之内发生分裂，要么永远不会发生分裂。（n是B的自动机节点个数）。
- 时间复杂度：$O(n^3)$

Makoto Soejima Contest 3
D. Subsequence

- 01串S有a个0和b个1，01串T有c个0和d个1，且T是S的子序列。问有序对(S,T)的方案数。

- a, b, c, d <= 2000

Solution

- 假设S和T都已经确定，现在要判断T是不是S的子序列，这个怎么做？

Solution

- 假设S和T都已经确定，现在要判断T是不是S的子序列，这个怎么做?
- 从前往后枚举T的每个字符c，从S中找到最小的i，使得S_i = c。然后从S中删掉$S_{1..i}$。

Solution

- 假设S和T都已经确定，现在要判断T是不是S的子序列，这个怎么做？

- 从前往后枚举T的每个字符c，从S中找到最小的i，使得S_i = c。然后从S中删掉$S_{1..i}$。

- 上述做法相当于找到字典序最小的$p_1,\ldots,p_k$,满足$T_i = S_{p_i}$。

Solution

- $p_1,\ldots,p_k$有一个性质：对于所有的$p_i < \mathrm{j} < p_{i+1}$，$S_j \mathrel{!=} S_{p_{i+1}}$。

Solution

- $p_1,\dots,p_k$有一个性质：对于所有的$p_i < \mathrm{j} < p_{i+1}$，$S_j \mathrel{!=} S_{p_{i+1}}$。

- 先构造一个字符串T，接着在T中添加字符构造字符串S，但是S要满足一个条件：T和S根据贪心求出的$p_1,\dots,p_k$这些位置恰好是原本T中的字符。

Solution

- $p_1,\dots,p_k$有一个性质：对于所有的p_i < j < p_{i+1}，S_j != $S_{p_{i+1}}$。
- 先构造一个字符串T，接着在T中添加字符构造字符串S，但是S要满足一个条件：T和S根据贪心求出的$p_1,\dots,p_k$这些位置恰好是原本T中的字符。
- 根据性质可知，所有字符’0’前只能加任意个’1’，所有字符’1’前只能加任意个’0’，末尾随意加。

Solution

- 枚举末尾需要加的'0'和'1'的数量，剩下的'0'可以加到任意一个'1'前面，剩下的'1'可以加到任意一个'0'前面。这里的方案只和T中'0'和'1'的数量有关，与T的具体状态无关，所以可以把T和S的方案分开来求。

2012 NEERC, Moscow Subregional Contest H. Hunt for Treasure!

- 给定一张n个点，m条边的有向图，要求添加n-m条边，使得满足图中每个点的入度和出度都为1且没有自环。假设每种合法的加边方式都是等概率的，问图中环的期望个数是多少。

- 0 <= m <= n <= 2000

Solution

- 如果只算方案呢?

Solution

- 如果只算方案呢?
- 第一步先把图中的环去掉，然后就剩下一些链和一些点，把这些链和点重新标号。

Solution

- 如果只算方案呢?
- 第一步先把图中的环去掉，然后就剩下一些链和一些点，把这些链和点重新标号。
- 为了保证不重复，先考虑标号最小的链：

Solution

- 如果只算方案呢?
- 第一步先把图中的环去掉，然后就剩下一些链和一些点，把这些链和点重新标号。
- 为了保证不重复，先考虑标号最小的链：
- 链尾向链头连一条边形成一个环，链的个数减一。

Solution

- 如果只算方案呢?
- 第一步先把图中的环去掉，然后就剩下一些链和一些点，把这些链和点重新标号。
- 为了保证不重复，先考虑标号最小的链：
- 链尾向链头连一条边形成一个环，链的个数减一。
- 链尾向另一条链的链头连边，链的个数减一。

Solution

- 如果只算方案呢?
- 第一步先把图中的环去掉，然后就剩下一些链和一些点，把这些链和点重新标号。
- 为了保证不重复，先考虑标号最小的链：
- 链尾向链头连一条边形成一个环，链的个数减一。
- 链尾向另一条链的链头连边，链的个数减一。
- 链尾向一个点连边，点的个数减一。

Solution

- 在没有链的情况下，考虑标号最小的点：

Solution

- 在没有链的情况下，考虑标号最小的点：
- 点向另外一个点连边，点的个数减二，链的个数加一。

Solution

- 在没有链的情况下，考虑标号最小的点：
- 点向另外一个点连边，点的个数减二，链的个数加一。

- 设$f_{i,j}$代表现在有i条链，j个点，形成若干个长度大于1的环的方案数，转移就按照前面几种情况一个个推就可以了。

Solution

- 在没有链的情况下，考虑标号最小的点：
- 点向另外一个点连边，点的个数减二，链的个数加一。
- 设$f_{i,j}$代表现在有i条链，j个点，形成若干个长度大于1的环的方案数，转移就按照前面几种情况一个个推就可以了。
- 设$F_{i,j}$代表现在有i条链，j个点，形成若干个长度大于1的环的所有方案下环的总数。转移和$f_{i,j}$类似，只需要在第一种情况的时候加一个方案数即可。

Solution

- 最后的答案就是$\frac{F_{i,j}}{f_{i,j}}$，但是还有一个问题：$f_{i,j}$和$F_{i,j}$可能会很大，会超过 long double的范围。

Solution

- 最后的答案就是$\frac{F_{i,j}}{f_{i,j}}$，但是还有一个问题：$f_{i,j}$和$F_{i,j}$可能会很大，会超过 long double的范围。
- n个点的情况下，根据容斥原理可知方案数$f_{0,n}$为：$\sum_{i=0}^{n}(-1)^i * \binom{n}{i}(n-i)!$，当n比较大时，$\frac{f_{0,n}}{n!} > 0.3$。在有链的情况下，比值只可能更大，所以考虑维护$\frac{F_{i,j}}{(i+j)!}$和$\frac{f_{i,j}}{(i+j)!}$。

Solution

- 最后的答案就是$\frac{F_{i,j}}{f_{i,j}}$，但是还有一个问题：$f_{i,j}$和$F_{i,j}$可能会很大，会超过 long double的范围。
- n个点的情况下，根据容斥原理可知方案数$f_{0,n}$为：$\sum_{i=0}^{n}(-1)^i * \binom{n}{i}(n-i)!$，当n比较大时，$\frac{f_{0,n}}{n!} > 0.3$。在有链的情况下，比值只可能更大，所以考虑维护$\frac{F_{i,j}}{(i+j)!}$和$\frac{f_{i,j}}{(i+j)!}$。
- 时间复杂度：$O((n-m)^2)$

SRM 615 LongLongTripDiv1

- 给定一张n个点m条边的无向带权图，问是否存在一条从1号点到达n号的长度为T的路径。

- $n, m \leq 50, T \leq 10^9, n$至少为2，边权不超过10000。

Solution

- 如果T比较小，那我们就可以记$f_{i,j}$代表从1号点出发到达i号点，并且路径长度为j是否可行。用bfs可以在$O(mT)$时间内可以算出答案。

Solution

- 如果T比较小，那我们就可以记$f_{i,j}$代表从1号点出发到达i号点，并且路径长度为j是否可行。用bfs可以在$O(mT)$时间内可以算出答案。

- 假设有一个长度为d的环经过1号点。如果从1号到n号点有一条长度为C的路径，那么一定有一条长度为$d*t+C$的路径(绕环走t圈)。

Solution

- 所以如果存在一条从1到n的路径长度为$C \le T$，且$C\%d \equiv T\%d$，那么就一定有一条从1到n的路径长度为T。

Solution

- 所以如果存在一条从1到n的路径长度为$C \le T$，且$C\%d \equiv T\%d$，那么就一定有一条从1到n的路径长度为T。
- 设$dis_{i,j}$代表从1号点出发到达i号点，路径长度为C，其中$C\%d = j$，最小的C是多少。这个可以在$O(md)$的时间内计算出来。

Solution

- 还有一个问题，就是如何找经过1号的环。

Solution

- 还有一个问题，就是如何找经过1号的环。

- 情况1：1号点的度是0，这种情况1号点是必然走不到n号点的。

Solution

- 还有一个问题，就是如何找经过1号的环。
- 情况1：1号点的度是0，这种情况1号点是必然走不到n号点的。
- 情况2：1号点的度不为0，假设有一条边(1,x,w)。因为图是无向的，所以就有一个长度为2w的环(1,x,1)了。$2w \le 20000$，所以能够在规定时间内算出答案。

SRM654 TwoEntrances

- 给定一棵n个节点的树以及s1和s2，现在要往节点上按顺序填入1,2..,n。如果路径(s1,y)或者路径(s2,y)上没有小于x的数字，那么节点y可以填x。问填数方案有多少种。

- n <= 3000

Solution

- 先考虑只有一个入口的情况。

Solution

- 先考虑只有一个入口的情况。

- 最大的数字一定填在根节点。子树之间是没有任何影响的，所以只要把这剩下的数字分配给子树，然后递归做子树的情况就好了。

Solution

- 设v_1=$s1$，v_2，…，v_k=$s2$是$s1$到$s2$的一条简单路径。

Solution

- 设v_1=$s1$，v_2，…，v_k=$s2$是$s1$到$s2$的一条简单路径。
- 如果节点v_{i+1}在节点v_i之前被数字占领，那么占领v_i的数字一定是由入口$s1$进来的。

Solution

- 设v_1=$s1$，v_2，...，v_k=$s2$是$s1$到$s2$的一条简单路径。
- 如果节点v_{i+1}在节点v_i之前被数字占领，那么占领v_i的数字一定是由入口$s1$进来的。
- 有一个看上去很对的做法：枚举$s1$到$s2$简单路径上的边，删掉它，然后变成两棵独立的树，然后分别计算方案。

Solution

- 这里有一个问题，就是简单路径上最先 被数字占领的节点v_i ，可以认为是从入口$s1$进来到达v_i，也可以认为从入口$s2$进来达到v_i。

Solution

- 这里有一个问题，就是简单路径上最先 被数字占领的节点v_i ，可以认为是从入口$s1$进来到达v_i，也可以认为从入口$s2$进来达到v_i。
- 假设，从$s1$和$s2$都能到达的点都默认由$s1$到达。

Solution

- 这里有一个问题，就是简单路径上最先 被数字占领的节点v_{i} ，可以认为是从入口$s1$进来到达v_{i}，也可以认为从入口$s2$进来达到v_{i}。
- 假设，从$s1$和$s2$都能到达的点都默认由$s1$到达。
- 设f_{i}代表节点$v_1, \dots, v_i$由$s1$到达，$v_{i+1}, \dots, v_k$由$s2$到达的方案数。g_{i}代表切掉边$(v_i，v_{i+1})$所得到的方案数。$g_{\mathrm{i}} = f_{\mathrm{i}} + f_{\mathrm{i+1}}$。

SRM642 WheelofFortune

- 有一个环，上面有n个数字，初始都为0。对环做m次操作，第i次操作定义为：从环上等概率选取一个长度为Si的区间，然后区间内数字都加一。
- 操作完后，环上有n个非负数字。有两个人A和B，他们事先不知道数字大小。A先取一个数字，被告知数字大小。然后B在知道A数字大小的情况下，再取一个数字。问两个数字之和的最大期望是多少。
- n,m <= 200

Solution

- 考虑第一个人的策略。因为是一个环，所以在不知道数字大小的情况下，环上每个节点都是等价的。也就是说第一个人随便取一个数字对答案的贡献都是相同的，假设第一个人取0号数字。

Solution

- 考虑第一个人的策略。因为是一个环，所以在不知道数字大小的情况下，环上每个节点都是等价的。也就是说第一个人随便取一个数字对答案的贡献都是相同的，假设第一个人取0号数字。

- 计算第一个人所取数字大小的期望。设$f_{i,j}$代表经过了i轮操作，0号数字大小为j的概率。$\sum_{j=0}^{m} f_{m,j} * j$就是第一个人的期望。

Solution

- 设g_j代表已知第一个人数字大小为j的情况下，第二个人能获得数字的最大期望。如果能计算出g_j,最终答案为：$\sum_{j=0}^{m} f_{m,j} * (j + g_j)$

Solution

- 设g_j代表已知第一个人数字大小为j的情况下，第二个人能获得数字的最大期望。如果能计算出g_j,最终答案为：$\sum_{j=0}^{m} f_{m,j} * (j + g_j)$
- 设$dp_{i,j,k}$代表经过了i轮操作，已知第0号数字大小为j，第k号数据的期望大小。

Solution

- 最暴力的转移：枚举第$i+1$轮的区间是哪个，进行转移。这样做的复杂度是$O(n^4)$的。

Solution

- 最暴力的转移：枚举第$i+1$轮的区间是哪个，进行转移。这样做的复杂度是$O(n^4)$的。
- 事实上，区间只有四种：覆盖了0号数字和k号数字的区间，覆盖了0号数字以及不覆盖k号数字的区间，不覆盖0号数字以及覆盖了k号数字的区间，不覆盖0号数字和k号数字的区间。

Solution

- 最暴力的转移：枚举第$i+1$轮的区间是哪个，进行转移。这样做的复杂度是$O(n^4)$的。
- 事实上，区间只有四种：覆盖了0号数字和k号数字的区间，覆盖了0号数字以及不覆盖k号数字的区间，不覆盖0号数字以及覆盖了k号数字的区间，不覆盖0号数字和k号数字的区间。
- 先枚举i和k，预处理出4种区间的个数，再枚举j进行dp转移，这样做就是$O(n^3)$的。

Solution

- g_j = max$\{dp_{m,j,k}$, 1 <= k < n$\}$
- $dp_{m,j,k}$是一个条件概率，转移的时候要注意一下。

谢谢大家！