2019 Multi-University Training Contest 2

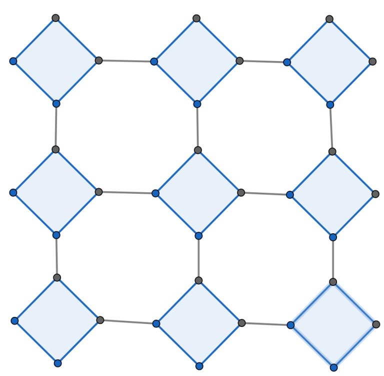
FZU Contest

Wednesday July 24, 2019

1 Another Chess Problem

问题相当于：给定一个棋盘，有一些格子上有障碍物，询问两个格子之间最短路以及方案数。

通过观察可以发现将每一个包含 4 个格子的正方形空地当作下图的一个蓝色菱形，下图的顶点对应表示棋盘的格子，这样就转化为在下图中计算。



可以先确定询问的两个点在这个图中是哪两个块，然后先考虑块间的移动顺序，可以发现连续走一个方向会比转方向多走 1 ，所以块间移动要使横着和竖着移动尽量错开，然后再考虑到点在块内也有个初始位置，所以可以枚举起点块出去的位置（顶点）和终点块进入的位置（顶点），最短路就是对四种情况取最小值。设每种情况最少连续同一个方向走的次数是 t ，最短路方案数 $=$ 块间移动的方案数 $\times 2^t \times$ 两个端点到出去方向的方案数。

块间移动的方案数可以用组合计数算出。

2 Beauty Of Unimodal Sequence

$f[i][0]$ 表示 $a[i]$ 一定取，序列 $a[1..i]$ 的最长上升子序列长度。

$f[i][1]$ 表示 $a[i]$ 一定取，序列 $a[1..i]$ 的最长单峰子序列长度。

$g[i][0]$ 表示 $a[i]$ 一定取，序列 $a[i..n]$ 的最长下降子序列长度。

$g[i][1]$ 表示 $a[i]$ 一定取，序列 $a[i..n]$ 的最长单峰子序列长度。

转移式子挺容易想的，留给读者思考。

枚举单峰子序列最高点下标，可以很方便的求出最长单峰子序列长度。

接下来逐位确定字典序最大的子序列。经过仔细观察可以发现，将候选集合按照下标排序之后，它们的值是单调的。利用该性质即可得出字典序最大最小的最长单峰子序列。

3 Coefficient

$$f(x) = \frac{b}{c + e^{ax+d}}$$

容易想到把分母乘到左边再双端求导得：

$$(c + e^{ax+d})f(x) = b$$

$$ae^{ax+d}f(x) + (c + e^{ax+d})f'(x) = 0$$

$$f'(x) = -af(x)(1 - \frac{c}{c + e^{ax+d}})$$

$$f'(x) = -\frac{a}{b}f(x)(b - cf(x))$$

假设这里a 暂时吃掉了b 并且取了相反数,得到：

$$f'(x) = af(x)(b - cf(x))$$

令X=f'(x)即有：

$$X' = aX(b - cX)$$

至此，可以看出X的任意阶导数都是关于X的多项式

且形如$X^{(n)} = XP_n(X), n \geq 0$

对上式再次求导，得到：

$$X^{(n+1)} = X'P_n(X) + XP_n'(X)X' = X'(P_n(X) + XP_n'(X))$$

$$= aX(b - cX)(P_n(X) + XP_n'(X)) = XP_{n+1}(X)$$

$$P_{n+1}(X) = a(b - cX)(P_n(X) + XP_n'(X))$$

以x代X,下面将多项式看成关于x的多项式：

$$P_{n+1} = a(b - cx)(xP_n)'$$

不难发现，上式可改写成：

$$P_{n+1} = (a(b - c\int)Dx) \circ P_n = (abDx - acx) \circ P_n$$

$$P_{n+1} = a(b(xP_n)' - cxP_n) = (\frac{abxP_n}{e^{\frac{c}{b}x}})'e^{\frac{c}{b}x}$$

假设这里a吐出了刚才吃掉的b：

$$\frac{P_{n+1}}{e^{\frac{c}{b}x}} = (\frac{axP_n}{e^{\frac{c}{b}}x})'$$

令$B_n = \frac{P_n}{e^{\frac{c}{b}x}}$带入得到：

$$B_{n+1} = (axB_n)' = aDx \circ B_n$$

故显然有：

$$B_n = a^n(Dx)^n \circ B_0$$

亦即：

$$P_n = (a^n(Dx)^n \circ (e^{-\frac{c}{b}x}))e^{\frac{c}{b}x}$$

令$A_k = \{a^n(k+1)^n(-\frac{c}{b})^k\frac{1}{k!}\}$

$$B_k = \{(\frac{c}{b})^k\frac{1}{k!}\}$$

$P_n = A \times B, P_n$第k项乘k!

注意这里 $\times$ 表示序列卷积

得到P_n之后，答案$ret = \frac{1}{n!}XP_n(X), X = f(x0)$

但是这只是回答了一个询问，显然不能每个询问都去做卷积

不同询问变化的参数是$a, b, c, d,$不变的参数是$n,$考虑变化的参数的影响

不难发现，a对答案的影响就是a^n, b的影响就是乘b, d显然没有影响

考虑c的影响，首先$X = f(x0)$发生变化，其次P_n中x^k这一项系数乘c^k

不妨令$a = b = c = d = 1,$然后得到P_n之后，在逐个询问进行微观调整

易得：ans=$\dfrac{1}{n!}\dfrac{a^n b}{c+1}P_n(\dfrac{c}{c+1})$

然后主要任务就转化为对已知的多项式P_n进行多点求值

而这是多项式的经典问题，可以使用时间复杂度为$O(nlog^2n)$的算法

4 Double Tree

首先对第一棵树进行边分治，假设当前我们正在考虑经过中心边 (st, ed) 的所有路径，我们不妨把切掉中心边之后所有和 st 联通的点标成黑色，所有和 ed 联通的点标成白色。

定义黑点 u 的权值 $h(u) = T_1.dis(u, st) + T_1.val(st, ed)/2 + val(u)$

定义白点 v 的权值 $h(v) = T_1.dis(v, ed) + T_1.val(st, ed)/2 + val(v)$

那么:

$$T_1.dis(u, v) + T_2.dis(u, v) + val(u) + val(v)$$

$$= T_1.dis(u, st) + T_1.dis(ed, v) + T_1.val(st, ed) + val(u) + val(v) + T_2.dis(u, v)$$

$$= h(u) + h(v) + T_2.dis(u, v)$$

现在对于边分治的每个联通块，我们需要考虑第二棵树。第二棵树上有些点是白色，有些点是黑色，有些点无色，对于每次修改，我们需要找一个黑点 u，一个白点 v 使得 $h(u) + h(v) + T_2.dis(u, v)$ 最大。

首先我们有一个结论：

对于一棵边权全是正的树，假如这棵树上有一个点集 A 的最长路端点分别是 u, v，另有一个点集 B 的最长路端点分别是 a, b，那么点集 $A \cup B$ 的最长路端点u, v, a, b 。

因为有修改操作，所以 $h(i)$ 的值是在动态变化的，我们用四元组 (i, l, r, w) 表示 i 点在时刻 $[l, r]$ 的权值 $h(i) = w$。对其进行线段树分治，则修改操作就变成了只有加边操作。

5 Everything Is Generated In Equal Probability

考虑一个长度为n 的随机排列（无相同元素），它所含逆序对数量的数学期望为$\binom{n}{2}/2$ 。因为每对下标对期望的贡献为1/2 ，且期望具有可加性。

容易得到：

$$f(i) = \frac{1}{2^i} \sum_{j=0}^{i} \binom{i}{j} f(j)$$

改写为：

$$f(i) = \frac{1}{2^i - 1} \sum_{j=0}^{i-1} \binom{i}{j} f(j)$$

这样可以$O(N^2)$ 预处理出$f(i), i \in [1, N]$ 。

$ans(n) = \frac{1}{n} \sum_{i=1}^{n} f(i)$ 。

$O(1)$ 回答每组数据。

时间复杂度$O(N^2 + Q)$ 。

6 Fantastic Magic Cube

一共有n^3 个单位立方体，将每两个不同单位立方体之间连一条边，边权为这两个单位立方体的价值乘积。如果将一个块A 切成块B 和块C ，显然就切断了B 与C 的联系，获得了它们之间的边权之和，因此无论怎么切，其实答案是一样的。

令$N = n^3$，a_i 表示第i 个单位立方体的价值。$ans = \sum_{1 \le i < j \le N} a_i * a_j = \frac{(\sum_{i=1}^{N} a_i)^2 - \sum_{i=1}^{N} a_i^2}{2}$。使用$fwt$ 计算出所有单位立方体值得分布（也就是计算每种值各有多少个）即可解出该式。

7 Game

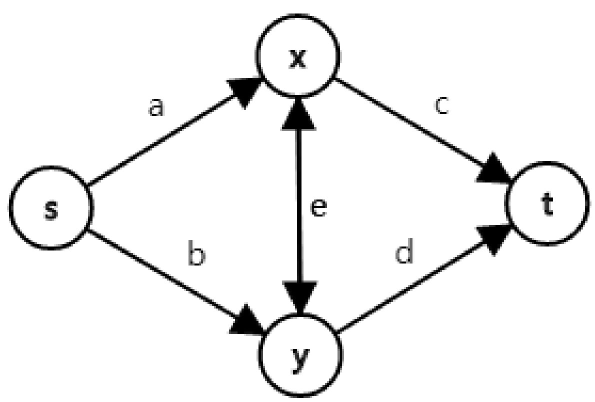
这是一个不平等博弈游戏，采用$surrealnubmer$ 计算游戏局面，最后计算游戏和的状态。

注意到游戏局面，没有超出$surrealnubmer$ 的表示范围；

如果计算出来的数> 0，左边的人获胜；< 0 右边的人获胜；$= 0$ 表示后手获胜；注意没有先手获胜的情况；更没有其他情况。

8 Harmonious Army

对每个士兵建立一个点x，点x 向源点s 连一条边，向汇点t 连一条边，分别表示选择两种职业，然后就可以先加上所有的贡献，通过两点关系用最小割建模，如下图所示。



设一条边的三种贡献为A, B, C，可以得到以下方程：

$a + b = A + B$（x, y 都选Mage）

$c + d = C + B$（x, y 都选Warrior）

$a + d + e = A + C$（x 选Mage, y 选Warrior ）

$b + c + e = A + C$（x 选Warrior, y 选Mage ）

可得一组解$a = b = (A+B)/2, c = d = (C+B)/2, e = -B+(A+C)/2$ ，然后将所有有关系的两点的图合并，用所有贡献减掉这个图的最小割即可。

9 I Love Palindrome String

求出本质不同回文串的数量分布（求每种回文串的个数），然后对每种快速$check$ 一下，叠加答案即可；可以用$manacher$ ，后缀自动机，回文自动机，字符串$hash$ 多种做法实现。

10 Just Skip The Problem

最优的方案必然是每次询问一个位的具体值，一共有n 个二进制位，方案数显然为$n!$。

复杂度$O(min(n, P)), P = 1e6 + 3$ 。

11 Keen On Everything But Triangle

首先考虑区间最大的三个数能否形成三角形，如果不能，考虑区间第二大、第三大、第四大的三个数，以此类推，直到能形成三角形。由三角形最小的两条边大于第三边的性质可知，只需要考虑区间的前44 大数即可(最坏情况下区间前几大数形成了斐波那契数列)。

时间复杂度$O(nlog_2n * 44)$ 。

12 Longest Subarray

如果右端点固定，对于每种元素，可行的左端点下标是两段连续的区间。

对于每种元素，将它的可行左端点区间在线段树中加一。

当右端点右移的时候，维护C 种元素的可行左端点。

查询时只需要询问线段树中最小的、值为C 的下标即可。