Day12 比赛题解

签到出得比较慢导致中期题都变成了后期题，后期题变成了送命题...

A. Wiki with Numbers

First Blood: 王必成 **4:12 (+)**

Author: rhy

很容易发现，$a_i + a_j - |i - j|$ 这个式子对于 (i, j) 和 (j, i) 都是完全一样的，也就是说我们可以只统计不大于 i 的 j 。于是我们就拆掉了假的绝对值把式子变为了 $a_j + j \geq k - a_i + i$ 。

从左往右枚举 i ，在值域上开线段树/树状数组就可以统计出空间在 $[1, i]$ 里，值域在 $[k - a_i + i, 2 \times n]$ 里的数的个数，累加进答案就做完了。

单组复杂度 $O(n \times \log n)$ 。

B. Wiki with R&C

First Blood: 张峻珲 **3:01 (+)**

Author: lst

求出圆心到矩形的最短距离 L 和圆心到矩形的最长距离 R 。

最短距离： 圆心到边上点的距离的最小值。

最长距离：圆心到四个点距离的最大值。

如果 $L > r$ (r 为圆半径),圆肯定与矩形不相交。

如果 $R < r$,圆包含了矩形,依然与矩形不相交。

如果 $L \leq r$ 且 $R \geq r$,那么圆肯定与矩形相交。

C. Wiki with Lucky Number

First Blood: 葛昊宇 **0:40 (+)**

Author: zcq

算法：思维+打表

这里可以以二叉树的形式将所有在范围内的幸运数字存在数组里，因为题中的数据是1000 000 000 ，所以最大的幸运数就是 5555 555 555 ，可知该树的深度为 11，因为第一层设成 0，而数组是从下标为 0 开始的，所以实际上就是去掉一个 0 ，多加一个最大幸运数，所以节点个数相当于深度为 10 的节点个数。当将幸运数从小到大的打表出来之后，就可以遍历选取 l 到 r 的幸运数即可，注意数据得用 $long\ long\ int$ 。

D. Wiki with Removal

Author: lst

题目大意: 给你一个大小为 n 的数组，你可以删掉数组中的任意 m 个数，问你在删除 m 个数之后剩下的数组有多少种。(其中数组的数的大小 $\leq k$)

解法: 可以假设 $next(i,c)$ 为在第 i 位之后第一个出现 c 这个数的位置，这样在转移的过程中我们只要删除 $[i,next(i,c)]$ 这个区间内所有的数，就可以形成一个新的数组了。

再假设 $dp[i][j]$ 为前 i 位数中已经删除了 j 个数的方案数，这样就可以得到状态转移方程为 $dp[next(i,c)][j+next(i,c)-i-1]+=dp[i][j]$ 。

最后的答案便为: $\sum_{j=0}^{m} dp[n-j][m-j]$ 。

E. Wiki with cxk

Author: jxc

题意是 3 个人 ABC 要在图上聚到一起，AB 有至多 10 个出现的点，C 可能全图出现，概率均相等，求聚集所需距离的期望。 首先观察到 AB 的范围很小，且作为概率分母来说，一共只有 $100 \times n$ 种情况，因此只需要计算任意起点组合下，聚集所需的距离和。 由数据范围想到枚举 AB 的出发点，把代价统一进图中，再统计 C 在任一点下的最小距离。先预处理 A 从每一个可能的起点出发，到其他所有点的最短路，记为 $disA_{ij}$，同理 $disB_{ij}$，表示从第 i 个可能的节点出发，到第 j 个节点的距离。 枚举 AB 起点，记 $dis_j = disA_j + disB_j$ ，即可把 AB 到点 j 的代价统计进图中，之后再对这张新图跑最短路，dis 中所有的值即代表 C 的选取对答案的贡献。 因为边权为 1，所有的最短路都可以由 bfs 代替，可以达到更优的时间复杂度。

F. Wiki with McDoDo

Author: jxc

题意是有 n 个点，m 条可建立的边，每个节点可以自己开店，代价 c_i ，也可以通过花费代价建立边集中的边，使得能和自己开店的节点联通。 最小化图联通的代价，考点很清晰地可以想到生成树，但本题还有自己开店这一选择。本题的一个重要事实：自己开店的节点，与其他有店的节点联通的节点，本质没有区别。 把自己开店这一操作转换，假设有一个 0 号节点已经有一家店，并且与所有 n 个节点连着一条代价为 c_i 的边，则开店操作等价转换成了连边操作，对新图跑裸最小生成树即可。

G. Wiki with Alladin

First Blood: 葛昊宇 3:05 (+1)

Author: zcq

算法：贪心+优先队列维护

本题本质上是需要判断一下消灭 n 只怪鸟所需要耗费的最小体力值是否大于原始体力值，所以只要算出最小体力值就可以了，所以自然想到的是贪心算法：

每个能量球有两个属性：光明魔法值和消耗体力值，这两个需要固定下一个来才能找出最小的体力消耗。

既然是找最小的体力消耗，那就优先确定光明魔法值来就可以，对怪鸟的黑暗魔法值和光明魔法值进行从大到小排序，这样能消灭前边的怪鸟的能量球同样也能消灭后边的怪鸟，用一个优先队列先输出耗费体力小的能量球，计算总的耗费体力值，然后和原始值判断一下就可以了。

特殊情况：如果用完这 m 只能量球都无法消灭这 n 只怪鸟则输出"No"。

H. Wiki With RPG

First Blood: 王必成 **5:29 (+5)**

Author: zcq

算法：单调队列优化DP

设 $dp[i]$ 表示 i 必须选时最小代价。

初值： $dp[0]=0, dp[1..n]=\infty$

方程： $dp[i]=min(dp[j])+in[i]\ and\ max(0,i-m)\le j<i$

为什么 j 有这样的范围？如果 j 能更小，那么 $[j,i]$ 这段区间中将有不符合条件的子区间，就会错。应保证不能有缝隙。 最后在 $dp[n-m+1..n]$ 中取最小值即答案，时间复杂度$O(n+m)$。

I. Wiki with Strings

Author: lst

考虑到总体的状态数只有 2^m 种，如果我们把某个串改变 1 位，这个改变后的串和这个串的答案就是 $m-1$，由此可见，每个串到另一个串都有一个距离，我们把这个距离设为改变的位数，所有串到某个串都有一个最小位数x，这个最小位数的最大值 y 的答案 $m-y$ 就是要的答案。

我们发现这就是一个最短路，直接拿 bfs 实现，复杂度 $O(n+m*2^m)$ 。

J. Wiki with Random Number Generator

First Blood: 张峻珲 **0:49 (+1)**

Author: rhy

原本这个题是求 $rand()\times rand()\ mod\ m=0$ 的概率的，出完发现做法假掉了，于是换成了加号。

由于逆元和快速幂都告诉大家了，签到应该不难。

枚举一个加数 i ，要使 $i+j\equiv 0\ (mod\ m)$ ，那么 $j\equiv m-i\ (mod\ m)$ 。

令 $g=-i\%m=(m-i\%m)\%m$ ，也就是说 g 是最小且非负的 j 。问题变成了有多少个数在区间 $[0,k-1]$ 里且等于若干个 m 加上 g 。

变形一下，问题等价于求 $[-g,k-1-g]$ 里有多少个 m 的倍数。

听说 cpp 的整除向 0 取整？我们给左右端点都加上 m 显然不会影响答案，那么就是求 $[m-g, k-1+m-g]$ 里有多少个 m 的倍数。i 的贡献就是 $\lfloor \frac{k-1+m-g}{m} \rfloor - \lfloor \frac{m-g-1}{m} \rfloor$ 。

单组复杂度 $O(k)$ 。

K. Wiki with fAKe Algorithm Generator

First Blood: 阙寅清 **3:26 (+)**

Author: rhy

受到多校 $exkmp$ 的启发，本来想出求求两两串之间 LCP 的暴力做法的比较次数的。后来发现和某一道原题很像就改成签到了。

令 $pre[i]$ 表示第 i 个字符和它左侧包括它自己在内的第一个 1 之间的距离。

令 $suf[i]$ 表示第 i 个字符和它右侧包括它自己在内的第一个 1 之间的距离。

这个显然正反扫一遍 $O(n)$ 就能处理完。

考虑第 i 个字符对答案的贡献，首先左侧肯定会比较完。

其次如果 $suf[i] \geq pre[i]$ ，那么右侧只会被比较 $pre[i]$ 次，对答案的贡献就是 $2 \times pre[i] + 1$ 。

否则，右侧只会比较 $suf[i] + 1$ 次，对答案的贡献就是 $pre[i] + suf[i] + 2$ 。

单组复杂度 $O(|s|)$ 。

L. Wiki with HEI!

First Blood: 王必成 **0:44 (+)**

Author: jxc

题意是一张 n 至多 100 的有向图，有一个点没有出边，但入边是满的，其他点没有限制。即若 x 是本体，"? x y"恒为 0，"? y x"恒为 1。使用不超过 128 次询问确定这个特殊的点。 直接从单次询问能获得的信息考虑，"? x y"这个询问，表示 $x \to y$ 这条边是否存在，如果存在，那么 x 不是本体，如果不存在，那么 y 不是本体。每次询问都可以保证排除一个点，通过 $n-1$ 次询问即可唯一确定本体。