

2023河南萌新联赛第（六）场：河南理工大学 题解

2023河南萌新联赛第（六）场：河南理工大学 题解

比赛链接：2023河南萌新联赛第（六）场：河南理工大学

A-简单的异或

B-这是dp题吗

C-旅游

D-买饼干的小Y

E-不爱吃早饭的小Y

F-爱睡大觉的小C

G-小H的人脉

H-左右横跳

I-简单的组合

J-线代高手

K-神运水晶

L-阴晴不定的大橘学长

感谢 河南工业大学 HaPpY1213 帮忙验题



比赛链接：[2023河南萌新联赛第（六）场：河南理工大学](#)

A-简单的异或

题意就是求一个 x ，使得区间内每一个元素与 x 异或后的总和最大

先考虑单个元素与 x 异或后最大，根据异或的性质，容易得到 x 应该和 a 的每一位都相反，才能使得异或值最大。

考虑区间大小大于等于2，假如区间内每个元素的二进制第 i 位相同(从右往左数)，比如全为1，那么当 x 的第 i 位取相反值 0 时，该位对于最后的总和贡献就是 $n * 2^{i-1}$ (n 表示元素的个数)，每有一个元素的二进制第 i 位为 0，该位的贡献就会减少 2^{i-1} ，当有超过 $n/2$ 的元素该位上为 0 时， x 的该位上取 1 贡献更大

所以贪心考虑 x 上每一位取值 0/1 的贡献的多少，选择贡献大的即可

区间内第 i 位上 1 的个数与 0 的个数，当 1 多时 x 的第 i 位上取 0，0 多则取 1。

特别的当 1 和 0 的数量相同时，因为题目要求输出较小值，取 0 即可

暴力计算区间内第 i 位上的 1 个数求解复杂度 $O(n^2)$

利用前缀和优化求解复杂度 $O(n)$

出题人代码链接: <https://www.luogu.com.cn/paste/ung8xbfi>

B-这是dp题吗

题目思路:

先假设最后一层的中间点是 mid , 那么根据题意往左或往右走次数之差不超过 k 次, 那么走到最后一层, 他的落脚点一定在 $[mid - k, mid + k]$ 之间, 那么问的就是走到最后一层在区间 $[mid - k, mid + k]$ 之间最大值是多少。

假设状态方程是 $f[i][j]$, 第 i 层在 j 这个位置所取得的最大值, 根据题意可以往下走, 往右下和左下走, 那么状态方程转移为

$$f[i+1][j+1] = \max(f[i+1][j+1], f[i][j] + a[i+1][j+1])$$

$$f[i+1][j] = \max(f[i+1][j], f[i][j] + a[i+1][j])$$

$$f[i+1][j-1] = \max(f[i+1][j-1], f[i][j] + a[i+1][j-1])$$

最后的答案就是在取最大值 $f[n][x]$, $mid - k \leq x \leq mid + k$

出题人代码链接: <https://www.luogu.com.cn/paste/ze2pcx5g>

C-旅游

题目思路:

很明显题中给的是一棵树, 题目中要求的是把每一个节点作为出发点得到的期望值中的最大值, 所以可以使用换根DP。

我们首先先求一个节点作为出发点的期望值, 定义 $f(x)$ 为以 x 为根节点的子树所能得到的期望值, 容易想到状态转移方程 $f(x) = \sum_{i=1}^{son[x]} \frac{f(son_i) + 1}{son[x]}$, 其中 $son[x]$, son_i 分别是节点 x 的孩子数量, 和第 i 个孩子节点。这样我们就可以求得以某一节点为出发点的期望值。

然后使用换根DP求每个节点作为出发点的期望值。定义 $dp[x]$ 为以 x 为出发点所得期望值, $son[x]$ 为点 x 的孩子数量, $siz[x]$ 为以 x 为根节点的子树走各个分支所得路程之和, 即 $\sum_{i=1}^{son[x]} f(son_i) + 1$ 。

状态转移方程可以这样考虑 $dp[j] = \frac{\text{各个分支路程之和}}{\text{孩子个数}}$ 。把根节点从 u 换到 j , 然后 u 就变成了 j 的一个分支, 那么这个分支的路程就是 $\frac{siz[u] - f[j] - 1}{son[u] - 1} + 1$, 分支总路程就是 $(siz[j] + \frac{siz[u] - f[j] - 1}{son[u] - 1} + 1)$, 然后孩子个数要+1, 就是 $son[j] + 1$ 。

所以状态转移方程即为

$$dp[j] = \frac{(siz[j] + \frac{siz[u] - f[j] - 1}{son[u] - 1} + 1)}{son[j] + 1}。$$

状态转移完成后, j 成为了根节点, 此时 $siz[j]$ 与 $son[j]$ 均发生变化, 所以要更新 $siz[j]$ 和 $son[j]$, 即:

$$siz[j] + = \frac{siz[u] - f[j] - 1}{son[u] - 1} + 1, \quad son[j] + =$$

当然由于 $son[u] - 1$ 可能为0, 所以我们还需要特判一下, 如果 $son[u] - 1 == 0$ 时, $dp[j] = \frac{siz[j] + 1}{son[j] + 1}$, 此时 $siz[j]$, $son[j]$ 分别更新为 $siz[j] + 1$, $son[j] + 1$ 。

出题人代码链接: <https://www.luogu.com.cn/paste/q9pxg5vf>

D-买饼干的小Y

题目思路:

先暴力依次让力气不为1的人去拿饼干，能拿完则答案就是去拿饼干的人数，若没拿完，则剩下的每一个饼干都需要一个人拿，总答案就是体力不为1的人数加上剩余饼干的数量。

出题人代码链接: <https://www.luogu.com.cn/paste/layuord4>

E-不爱吃早饭的小Y

题目思路:

本题仅考查简单的SG函数的套用。对于单堆饼干暴力或者递归求出该堆的SG函数值，如果为0，先手必败，反之先手必胜。对于n堆游戏，总游戏的合就是每个单堆的SG函数的异或和。

$$ans = SG_1 \oplus SG_2 \oplus \dots \oplus SG_n$$

出题人代码链接: <https://www.luogu.com.cn/paste/kjzzxo85>

F-爱睡大觉的小C

题目思路:

期望 = (全部第k大数的和)/(区间个数)

显然区间个数是 $(n - k + 2) * (n - k + 1) / 2$ ，所以只需计算所有长度大于k的区间的第k大的数的和即可。

反过来考虑每个数作为第k个大数的区间，只需要计算出每个数作为第k大的数的区间个数就能算出答案

区间个数需要不重复，不遗漏的计算，我们考虑从最小的数开始计算，只需要包含这个最小的数，长度为k的区间就行，换句话说就是找这个数位置上的左边和右边，一共找到k-1个数。那这些长度为k的区间的第k个大数一定是当前的最小值。那剩下的区间一定不是以当前的最小值作为第k大的数了，然计算次小值，但当次小值遇到了前面的最小值就需要再多往左/右取一位，为了减少跳过的时间，我们可以考虑算完一个最小值就删除一个最小值，然后仍然找长度为k的区间。

那有没有可以实现连续位置的删除操作的数据结构那，当然有了！就是我们的链表，由于数据范围很大，我们可以先排序所有数字，再从最小值开始，在链表上找左右的区间，每次算完直接把最小值删除即可，这样一来时间复杂度就变成了 $O(n * q)$

出题人代码链接: <https://www.luogu.com.cn/paste/nahik6ba>

G-小Hの人脉

题目思路:

首先可以用LCA求出每个人从出发点到目的地花费的总金钱数，记录下来。

因为题目要求的是花费钱数最多的人最少花费了多少元，所以可以二分找到该答案。

在二分的check函数中枚举所有人的总金钱数，找出比二分的钱数多的路径，对该路径进行树上差分标记出来，并记录这样的路径个数为cnt。

进行dfs统计每条边被标记了几次，如果某条边被标记的次数等于cnt，那就说明这条边被所有大于二分的钱数的路径都经过了。因为要将一条边免费，所以我们肯定要免费一条被所有大于二分的钱数的路径都经过的边，且这条边的边权被减去后，可以使所有人的路径的总金钱数都小于等于二分值，那么本次check函数才可以返回1。否则，返回0。

在check的同时，我们可以用一个变量 num 表示被免费的那条边的边权值。那么最终答案就是二分值和 num 。

出题人代码链接: <https://www.luogu.com.cn/paste/ar13j6pf>

H-左右横跳

题目思路：

简单的动态规划，题目可以理解为一个 $n*m$ 大小的点阵，每个点上有不同的分数，而一个点上能得到的分数或积累的分数需要通过操作一直接继承正下方的点的分值，或者通过操作二继承下方任意一列的间隔 k 行的点的分数并加上所在位置上的分数，那么即可推出如下状态转移方程：

$$f_{i,j} = \max(f_{i-1,j}, \max(f_{i-k,1}, f_{i-k,2}, \dots, f_{i-k,m}) + w_{i,j})$$

另外，因为通过操作二转移得到的分数需要比较 m 列所有的分数，所以每操作一行就记录当前一行所能获得的分数的最大值 $maxv$ ，即：

$$maxv_i = \max(f_{i,1}, f_{i,2}, \dots, f_{i,m})$$

$$f_{i,j} = \max(f_{i-1,j}, maxv_{i-k} + w_{i,j})$$

最后注意边界情况和初始化即可。

出题人代码链接: <https://www.luogu.com.cn/paste/12c2r5b5>

I-简单的组合

题目思路：

将一个小于 2^{32} 的32位二进制无符号数每8位分成一部分，则每一部分都可以表示成一个小于256的数，将这4个数从小到大排序，从小到大依次乘于 $2^0, 2^8, 2^{16}, 2^{24}$ 然后相加即是答案。

出题人代码链接: <https://www.luogu.com.cn/paste/3kblggwp>

J-线代高手

题目思路

数据范围较小，直接暴力枚举左上角和右下角，如果该子矩阵合法，更新答案即可。

要注意的是如果暴力的遍历子矩阵的每一个元素来求其元素和的时间复杂度为 $O(nm)$ ，这样总的时间复杂度为 $O(n^3m^3)$ ，无法通过此题。因此需要用前缀和来快速求一个子矩阵的元素和，总时间复杂度为 $O(n^2m^2)$ 。

出题人代码链接: <https://www.luogu.com.cn/paste/rco8szhc>

K-神运水晶

题目思路：

先不考虑选择0的情况：

考虑 dp ，令 $dp[i][j]$ 表示选出和为 i ，且最后一次选择了 j 的序列的概率，转移为：

$$dp[i][j] = \sum_{l=1}^j dp[i-j][l] \times \frac{1}{k}$$

可以暴力枚举 l 转移，也可以用前缀和优化为 $O(n^2)$

由于序列非递减，所以0只会出现在序列的前缀。假设选出没有0的合法序列概率为 p ，通过枚举0的个数，可得到答案：

$$ans = (1 + \frac{1}{k} + \frac{1}{k^2} + \frac{1}{k^3} + \dots) \times p$$

该式可以利用等比数列求和公式求得。

出题人代码链接：<https://www.luogu.com.cn/paste/1diyd06w>

L-阴晴不定的大橘学长

题目思路：

对于区间和 $\sum_{i=l}^r a[i] \geq x$ 我们可以使用前缀和来将公式转化为 $sum[r] - sum[l-1] \geq x$ ，如果我们想要快速查找以 r 结尾的区间有多少满足条件的区间，那么公式可进一步转化为 $sum[r] - x \geq sum[l-1]$ ，这时只要找出在 r 的左边小于等于 $sum[r] - x$ 的区间前缀和数量即可，由于会有负数的出现，前缀和不是递增的，所以我们可以使用树状数组来维护 r 的左边小于等于 $sum[r] - x$ 的前缀和数量，统计 r 左边小于等于 $sum[r] - x$ 的前缀和出现的次数，同时，前缀和的值可能很大，所以我们要使用离散化，将数值范围降至数组可以容纳的范围，查询的复杂度是 $O(\log^2 n)$ ，总的时间复杂度是 $O(n \log^2 n)$ 。

出题人代码链接：<https://www.luogu.com.cn/paste/o2xon9fx>

感谢 河南工业大学 [HaPpY1213](#) 帮忙验题

thanks for reading

--河狸算法协会