

Problem A. osu!

对于每个s，对于答案的贡献是它左边的o的数量*它右边u的数量，o和s的数量分别用前缀和和后缀和可以很轻松的 $O(n)$ 预处理。

Problem D. 史莱姆

特判 $n=1$ 的情况，剩下的最终史莱姆的生命值一定均为素数，即最终数量为 n 的质因子数量。

Problem E. 比比谁更大

99068070是一个特殊的模数， $99068070=2*3*5*7*11*13*17*19*103$ 。对于一个给定的 n ，求 $n! \% 99068070$ 时；由阶乘的定义可知，当 n 大于等于103时，阶乘结果中将包含模数99068070中的所有质因子，此时 $n!$ 的结果可以整除模数99068070，由%的定义可知此时 $n! \% 99068070=0$ 。因此，在计算 $n! \% 99068070$ 时，当 n 大于等于103时，答案为0； n 小于103时，暴力计算即可。用这种方法算出a、b各自的结果，然后比较出结果即可。

Problem F. 拼牛牛

函数收敛性相关 可以明显看出增长数字是一个等比数列，根据等比数列的求和公式知道 $(1 - k^n)/(1 - k)$ 在 k 小于1大于0时，函数收敛，极限值是 $1/(1 - k)$ 因此只要比较这个极限是否小于等于当前还差的钱数，如果小于等于还差的钱数则一定不能够提现，否则一定可以提现。

Problem G. 歪脖子树下的灯

方法一：二项分布

由于每次开关都不会相互影响，那么这题符合 n 次独立重复的伯努利试验。所以可以使用二项分布公式来计算，但是由于此题的特殊性，不能全取，只能取奇数项。

所以答案为：

$$ans = \sum_{k=1}^n C_n^k p^k (1-p)^{n-k} (k : Gp)$$

化简一下式子：

$$ans = \frac{1 - (1 - 2p)^n}{2}$$

方法二：递推。

设 $f[i]$ 为第 i 次拉后，灯为亮着的概率。已知 $f[0]=0$ ，求 $f[i]$ 。

$$f[i] = (1-p) * f[i-1] + p(1-f[i-1])$$

$$= (1-2p) * f[i-1] + p$$

Problem H. 水题

考虑把时间花费分成三部分计算。

1. 水的装与卸的时间贡献。每一桶水的装卸时间花费是2，不会因为送水顺序而改变答案的大小。

$$t_1 = \sum_{i=1}^n a_i * 2$$

2. 送每桶水的时间贡献。每一桶水被送到第 $i+1$ 层楼所需要的花费是 i （无论这桶水是在何时，通过哪次运输，贡献都是固定的）。

$$t_2 = \sum_{i=1}^n i * a_i$$

3.送水车的时间贡献。使用贪心的策略，每次尽可能的优先满足最高楼层的需求。如有剩余，则满足次高楼层的需求，一直延续下去，直至送完所有水或者达到k桶。

注意：送水车的轨迹是一个来回，故时间贡献需要乘2。

解析如下：

设当前最高楼层为 $i+1$ 层，需要送 x 桶水，送水车每次最多装载 k 桶水。

当 $x > k$ 时，为了满足该楼层的送水需求，必须要送 x/k (向下取整) 次。

当 $x \leq k$ 时，按照上述策略，则先把该楼层所需要的 x 桶先送达。如果不把所需要的 x 桶送达，而留下大于等于1桶水，那么下次送水必须要回到当前最高楼层，这样必定会增加时间花费。

用 $O(n)$ 的方式，从最高楼层往低楼层计算一下送水车的时间贡献，记为 t_3

最终答案为：

$$ans = t_1 + t_2 + t_3$$

Problem I. 烧烤丝瓜

首先如果三哥进入厨房发现炉子关闭，一定要打开，故可以将三哥不断反复直到发现炉子关闭然后打开设为一个周期，可以求出一个周期中炉子开的时间和炉子关闭的时间，然后分剩下不足一周期的时间是炉子一直打开还是存在关闭的时间分类算即可

Problem J. 过生日

对于字符串中的每一种字符用桶记录他们在字符串中出现的位置(按顺序)，在每个桶里，用相距为 $k-1$ 的两个元素值之差的绝对值更新答案。复杂度 $O(n)$

Problem K. 区间交

将 m 个区间按照左端点进行排序，遍历所有端点，处理一个优先队列，存右端点。（最小值的时候用最大堆，最大值的时候用最小堆）当队列元素数量少于 k 时入队，当队首大于右端点的时候，将队首出队然后将这个区间的右端点入队，解就是最小的队列中最小值-区间左端点+1（最小值的解法，最大值同理）

Problem L. 排名

$rank$ 只与 cnt 值的大小有关， cnt 值相同的数， $cnt \ xor \ rank$ 相同，因此只需统计每种 cnt 值有多少个，又因为不同的 cnt 值最多只有 \sqrt{n} 种，可以暴力维护每种 cnt 值的个数，以及每种 cnt 值对应的 $rank$ 值。

不同的 cnt 值最多只有 \sqrt{n} 种说明：在极限情况下， $cnt = 1, 2, 3, \dots, x$ ，每种 cnt 值各有一个，此时满足 $1 + 2 + 3 + \dots + x \leq n$ ， x 的最大值是 \sqrt{n} 级别的。