

打怪

对于一个子集，花费的金币个数是这个集合中血量的最大值，而奖励的金币是这个集合中的不同的数字个数。

因为线性性，这两部分可以分别计算。

对于集合中的最大值，可以枚举最大值，然后直接计算有多少集合的最大值是它。

对于不同的数字这部分，可以枚举每个数值，然后计算这个数出现了至少一次的方案数即可。

时间复杂度 $O(n\log n)$ 。

与之国

发现整张图是一棵树，于是答案等于点数减边数。发现对于每个陆地 (x, y) ，如果不是 $(0, 0)$ ，则 $(x, y - 1)$ 和 $(x - 1, y)$ 恰有一块是陆地。

因此只需对子矩形的第一行和第一列计算答案即可。

对于只有一行的情况，仍然考虑点减边。

对于询问 (a, l, r) ，计算点数相当于计算 $\sum_{i=l}^r [a \text{ and } i = 0]$ ，计算边数相当于计算 $\sum_{i=l}^{r-1} [a \text{ and } (i \text{ or } (i + 1)) = 0]$ ，可以使用数位 dp 计算。

时间复杂度 $O(nm)$ 。

爬山

不失一般性的，假设 $a < b$ 。

可以发现，策略一定是先从 a 爬到 mid ，再从 mid 爬到 b ，其中 mid 是 a 到 b 中最高的山的高度。

显然两部分对称，接下来只考虑 a 到 mid 的情况。

由于向下（或左/右下）不优，向上（或左/右上）次数固定，于是可以得到策略：

1. 一直往右上走，如果被挡住则往上，直到高度达到 h_{mid} 。
2. 平飞到 (mid, h_{mid}) 。

显然在 a 到 mid 的过程中，较后且较矮的山不需要考虑。

数据随机下，需要考虑的山数量是 $O(\log n)$ 个的，预处理出每个山的后继即可。

考虑更严格的，如果山 i, j 满足： $i < j, h_i - i > h_j - j$ ，则山 j 不需要考虑。

否则这些山一定都会对小 L 产生阻挡。

发现总向上次数就是 $\max_{i \in [a, mid]} (h_i - i) - (h_a - a)$ ，于是可以计算答案。

每次询问只要求三次区间最值即可。

时间复杂度 $O((n + m) \log n) \sim O(n + m)$

星际战争

首先求出所有特殊点，特殊点的定义是从 1 开始做一遍 bfs 后，拥有非树边的点。这样的特殊点最多有 200 个。

接着我们把这些特殊点到所有点的距离都算出来。

然后我们枚举把基地建立在哪个星球上，对于每个特殊点，我们都可以算出来通过这个特殊点黑暗势力与光明势力的分界点，分界点下面的子树归光明势力所有（注意特殊边（非树边）连接的两个点的深度差最多是 1）。相当于我们有一些子树，要求出它们的并的大小，这个可以用 dfs 序列解决。