

## A、谜之礼物

---

难度：3

题型：BFS搜索+状态压缩

BFS搜索+状态压缩即可。难点在于合理的状态表示。 $dp[x1][y1][x2][y2][S]$ 表示目前已经走过的坐标集合和两个人对应的坐标的状态的最短时间。其中S需要使用二进制状态压缩进行存储。转移则需要枚举两个人的 $4 \times 4 = 16$ 种走法。这样的空间复杂度是 $O(2^{16} * 4^2 * 4^2) = O(2^{24})$ 。

## B、喜之游乐场

---

难度：2

题型：多重背包

这道题一眼看去是个01背包问题。但是N和M范围都在10万， $O(NM)$ 的DP没办法在时限内出解。

进一步分析我们可以发现Wi和Pi的范围都很小，在10以内，所有本质上不同的游戏最多有100种，每种可能游戏有很多个。(花费i元，获得j元的游戏有多个)于是这道题又变成了一道**多重背包问题**。不过多重背包如果直接DP仍然是 $O(NM)$ 的。我们使用“二进制分解”的优化方法。由于我们是把同类的C件物品分解成了 $O(\log C)$ 件物品。所以最大数据100种，总计10万件物品最多被分解成大约1000件物品。使用01背包的DP，复杂度 $O(1000M)$

## C、醉之见面会

---

难度：4

题型：莫队算法+树状数组

首先我们根据贪心可以知道，对于一个连续区间 $[l, r]$ 的小迷粉发言问题，一定是将用时短的排在前面、用时长的排在后面。(想证明可以用高中排序不等式)

于是我们有个暴力算法，就是对于每一个询问，将 $a[l] \dots a[r]$ 进行排序，然后直接计算结果。这样的复杂度是 $O(T * n * \log n * m)$ 的，会超时。

解决这道题目，我们需要使用到“莫队算法”。莫队算法对于这类区间上的询问问题，给出了一个解决的算法框架。剩下需要考虑的问题是，当我们增加一个数，或者减少一个数时，结果会如何变化。

比如当前区间从小到大排序后是 $B[1], B[2], \dots B[N]$ ，当新插入一个X时，假设排序后是 $B[1], B[2], \dots B[K-1], X, B[K], B[K+1], \dots B[N]$ 。

则 $B[1], B[2], \dots B[K-1]$ 等待时间没有增加。X等待时间是 $B[1] + B[2] + \dots + B[K-1] + X$ ，即一个前缀和+X。 $B[K], B[K+1], \dots B[N]$ 的等待时间都增加X。所以我们需要数据结构维护一个有序数组，知道新增加的元素在数组中排第几大，并且知道有序数组的前缀和。可以用树状数组搞定。

当删除一个数时，处理也是类似的，不再讨论。