Solution

Garden

算法一

直接枚举计算，时间复杂度 $O(n^2m^2k)$ 。

算法二

两个位置如果无交，那么贡献可以直接相加。

枚举一个位置的时候，与其有交的最多 k^2 个位置。

特殊考虑这些位置，时间复杂度 $O(nmk^3)$ 。

Inverse

算法一

枚举交换对，每次重新计算逆序对，时间复杂度 $O(n^3\log n)$ 。

算法二

看成二维平面上有 n 个点 (i,a_i) 。交换操作只会发生在逆序对身上，而其减小的逆序对数为两个点之间圈出来的矩形内的点数乘 2 再加上 1（两个点本身）。

于是可以枚举第一个点，第二个点用树状数组维护矩阵点数，时间复杂度 $O(n^2\log n)$ 。

算法三

观察点对 $i<j$ 选择了 (i,a_i) 和 (j,a_j) 。首先 i 必须是前缀最大值，如果有 $k<i,a_k>a_i$ 那么将 i 改为 k 不会变劣（矩形变大）。j 同理只能是后缀最小值。

再考虑一个点，他会对哪些 (i,j) 做出贡献？首先在他左上侧的 i，当我们只保留前缀最大值的 i 时，目标的 i 因当时一段区间。j 同理。因此一个点的贡献，实际上是限制了 i,j 分别位于两个区间内。

这样的化，将所有前缀最大值提取、后缀最小值提取，每个点的贡献即为矩形。我们要找的是矩形覆盖次数最多的 (i,j)，线段树扫描线即可。时间复杂度 $O(n\log n)$ 。

Walk

算法一

暴力枚举所有序列。时间复杂度 $O(nmS!)$ 。

算法二

一个经典的结论是，每条边被经过次数的上限为两侧权值和较小的值，而取带权重心后，在多个子树之间来回跳跃即可构造目标路径，也就是说我们完全能取到上界。

那么每次找到带权重心，以其为根，每条边只要求子树权值和，就是经过次数。

时间复杂度 $O(nm)$ 。

算法三

注意到每次修改其实只会影响 c,d 路径上的边的贡献，剩下的边因为两侧子树权值和并未发生变化，所以不用重新计算贡献。

用数据结构维护信息，每次询问独立，当树是完全二叉树时，每次只需要重新暴力计算 $O(\log n)$ 条边的贡献。

可以通过特殊性质 A 。

算法四

当树是链时，修改一条边后，树也只会是 "工" 字状。由于结构很简单，可以二分重心的位置。

然后用数据结构维护信息。

可以通过特殊性质 B 。

算法五

不妨把 c,d 在原树上的对应链提出来，加上 (c,d) 构成一个环。整棵树就变成了环套树。

把环以外的所有边的贡献先算出来，这个可以通过子树和的方式计算。现在考虑断掉 (a,b)，剩下这条链上的边，贡献是具有单调性的：只需要二分出中点即可。

用倍增实现，时间复杂度 $O((n+m)\log n)$ 。

BONUS：利用离线，可以得到复杂度更好的并查集做法。

Sail

算法一

输出 1，当 $n=1$ 时第一个时刻总能上岸。时间复杂度 $O(1)$ 。

算法二

对于 $p_i \in \{0,100\}$，问题弱化为普通图论问题。以 (i,v) 表示在位置 i，速度为 v 的状态，首先 $|v|$ 的级别是 $O(n)$ 的（否则肯定已经上岸），以此建立 $O(n^2)$ 个点。剩下的 DFS 即可。

算法三

先处理 -1（即将概率不为 0 的转移建图，然后缩点，处理所有出不去的点，拓扑排序倒推即可知道哪些起点是有概率走不出去的，那么答案就是 -1 了）。

我们对于所有剩下的合法状态 (i,v) 设计 DP：$dp_{i,v}$ 表示状态 (i,v) 走出去的期望步数。通过高斯消元可以求出所有 $dp_{i,0}$ 。

时间复杂度 $O(n^6)$ 。

算法四

注意当 $|v|$ 到达 $O(\sqrt{n})$ 级别的时候，一定走出数轴了（即 $1+2+\cdots+\sim 2\sqrt{n} \geq n$，如果能到达这个速度，则总位移已经不小于 n）。

因此 $|v|$ 实际上只要保留 $O(\sqrt{n})$ 级别的状态即可。

时间复杂度 $O(n^{4.5})$ 。

算法五

再次减少状态数量：我们只保留 $v=0$ 的状态。

对于 $i<j$，用 DP 计算出 $(i,0)$ 下一次遇到 $(?,0)$ 的状态是 $(j,0)$ 的 **期望步数**、**概率** 。首先 i 一定是一直往右走的，因为最后一定有 $v>0$，而若之间出现 $v<0$，由于 v 连续，所以一定到不了 j 就遇到另一个 $(?,0)$ 了。

所以转移不成环，我们完全可以通过 DP 计算上述信息。对每个 (k,v) 记录 $(i,0)$ 到达其的期望步数、概率即可，中间限制 $v\neq 0$，然后在 $(j,0)$ 处统计贡献。顺带记录 $(i,0)$ **直接走出去** 的期望步数、概率。

这一部分时间复杂度 $O(n^{2.5})$（同理 $|v|$ 是 $O(\sqrt{n})$ 的）。$i>j$ 也是同理。这样就可以得到所有 i,j: $(i,0)$ 下一步是 $(j,0)$ 的期望步数、概率了。

以上面的信息列关于 $(i,0)$ 的方程，就可以高斯消元解出所有 $(i,0)$ 的答案了。

总时间复杂度 $O(n^3)$ 。细节可以参考 std 。