solution

斜率(slope)：

对于 50%的数据：

直接枚举两点用斜率公式计算即可，时间复杂度 $O(n^2)$。

对于 100%的数据：

最大斜率一定在横坐标相邻的两个点产生。证明很容易，假设最大斜率两点间有一点，则这个点和两个点中至少一个点的斜率不小于这两点的斜率。因此我们只要按横坐标排序扫描一遍即可，时间复杂度 O(nlogn)。

最优路线(path):

对于 50%的数据：

枚举路径上最大点权，只保留点权不超过某个点的点，使用 floyd 计算每两个点之间的最短瓶颈路，时间复杂度 $O(n^4)$。

对于 100%的数据：

考虑 floyd 的本质：f[k][i][j]表示只经过前 k 个点的情况下 i 到 j 的最短路，一般按节点标号顺序枚举 k 即可求得两两最短路。这里我们可以改变枚举顺序，按点权从小到大枚举 k，这样 f[k][i][j]就是点权不超过第 k 个点，i 到 j 的最短路，在每个阶段都更新一遍两两点对答案，时间复杂度 $O(n^3)$。

小 G 的线段树(segment)：

对于 20%的数据：

暴力枚举所有可能的操作序列，使用期望的定义式计算每个询问的答案，时间复杂度 $O(n!*n^2)$。

对于 50%的数据：

容易发现，一个区间的和的期望，等于区间内所有点的答案的期望的和。因此我们可以求出每个点最后的期望答案，最后回答所有询问。对每个点，首先找出所有覆盖该点的操作。在最后一个赋值操作之前，所有操作都是无效的。每个赋值操作出现在最后的概率都是相等的，所以赋值操作对答案的贡献是它们的平均数。而对于加法操作，当且仅当这个操作出现在所有赋值操作之后才有贡献。即相当于 k+1 个物品随机排列，某个物品出现在最后的概率，为 1/(k+1)。这样对每个点计算完答案后求出前缀和，询问时只需用前缀和相减。时间复杂度 O(nm+q)。

对于 100%的数据：

上述对每个点答案的计算，可以使用差分解决，只需对赋值操作的和、加法操作的和、赋值操作的次数作差分，就能在 O(n+m)的时间内求得每个点的答案，时间复杂度 O(n+m+q)。