T1

对于30%的数据，直接$O(n^2)$计算即可。

注意对于每个点，计算公式为
$\sum(x_i-x_j)^2+(y_i-y_j)^2=\sum x_i^2+x_j^2+y_i^2+y_j^2-2x_ix_j-2y_iy_j$

只要预处理所有$\sum xi^2,\sum y_i^2,\sum xi,\sum y_i$即可对于每个点$O(1)$计算。

时间复杂度$O(n)$。

T2

注意到删除后的答案为第一大的石子+第三大的石子+第五大的石子....。

那么暴力直接枚举删去哪些石子即可。

那么按照石子大小排序后dp。

令$dp_{i,j}$表示前i个石子，删去j个后A的最大得分。

枚举第i个石子是否删去即可。

T3

显然暴力dp的复杂度是$O(Q^2a_i)$的。

考虑用线段树分治，记入每个石子堆加入的时间，然后进行dp，时间复杂度$O(QlogQa_i)$。期望得分40分。但是由于常数太过于优秀，实际获得了100分。

考虑把区间劈开，变成两半。每一半用一个栈维护栈的任意位置的任意一点走到栈底任意一点的贡献和。

那么在栈中加入一列的复杂度为$O(n)$，删除一列的复杂度为$O(1)$。

如果栈被删空了，直接取中间位置重新劈开，暴力重构。

可以通过势能分析证明复杂度为$O(Qn)$。

T4

考虑暴力，显然每次每个位置只会被修改$log(loga_i)$次。用并查集维护这个修改，时间复杂度$O(nq\alpha log(loga_i))$。

事实上，可以用前缀和统计每个位置被修改了多少次，然后预处理每个位置被处理x次的值。时间复杂度$O(nq)$。

事实上还有暴力是回滚莫队，但是复杂度没有得到证明。因为复杂度是均摊的。

按照l_1排序后，r_1期望最长上升子序列为$O(\sqrt{Q})$。

也就是可以划分为$O(\sqrt{Q})$个询问集合，每个询问集合的修改关系是嵌套关系。因为l_1是递增的，r_1是递减的。

注意到问题就变成了每次对一个区间开根，然后询问区间和。

那么每个位置只会被修改$O(log(loga_i))$次。用并查集维护修改。区间和可以用分块做到$O(1)-O(\sqrt{n})$时间复杂度
$O(q\sqrt{n}+n\sqrt{q}loglog(a_i)*\alpha)$

（n,m同阶）