Solution

AddEdge Loves Working

用一个堆维护未关电源的包间的使用结束时间，将事件按 s 排序，每次扫到一个事件，将时间差超过 m 的堆顶元素弹出。

LargeDumpling Loves Graph

20%：大概怎么写都可以。

50%：令 dist[i][j]表示当前在 i，已经走过 j 条边的最短路径，拆点跑最短路即可。

70%：实际上对于相同起始点的询问，dist 数组是相同的，那么我们将询问按起始点排序或者预处理出 n 个 dist 数组就可以 O(1)回答了。

100%：令 dist[t][i][j]表示从 i 开始走 2^t 条边到 j 的最短路径，类似 Floyd 的方法维护即可。由于对于所有询问的 e 是固定的，所以我们可以在求 dist 的同时维护出 ans[i][j]，即从 i 到 j 的答案。时间复杂度 $O(n^3 \log m)$。

jvjhfhg Loves Sequence

20%：大概怎么写都可以。

50%：我们注意到一个区间内第 k 大的数不小于 m 和一个区间内至少有 k 个数不小于 m 是等价的。我们将不小于 m 的数看作 1，小于 m 的数看做 0，于是问题转化为询问有多少个区间内 1 的个数大于等于 k。由于 k 很小，对于$\forall i \in [1,k]$我们可以使用线段树来维护左起、右起及区间内部 1 的个数不小于 i 的区间个数。使用类似[SCOI2010]序列操作的方法维护即可。

20%的 q=1 特殊数据：求区间内 1 的个数不小于 k 的区间个数是 two-pointer 经典问题，two-pointer 扫一扫就没了。

100%：将所有询问按右端点排序，使用一棵线段树维护每个点作为左端点的合法区间个数。每次加入一个数 a[R]，我们使用 two-pointer 同时维护出最靠右的合法左端点 L，同时在线段树上更新 1~L 的值，然后处理所有 r 在这个位置的询问。由于当前已计数的合法区间的右端点均在 R 或 R 左侧，且区间不会存在左端点大于右端点的情况，所以我们在线段树上直接询问[l, r]即可。