a.选座

算法1

首先可以枚举旅客选座顺序，枚举每个旅客选啥，暴力搜索此题。
复杂度：$0(？)$
期望得分：15-25分

算法2-1

考虑第一问，发现对于一个旅客，考虑如果它最后一个选座，那么假设它和 c 个旅客的行程有相交，那么至少需要 $c+1$ 个座位。那么枚举每个旅客找出有多少个和它相交，取个需要的最大值就行了。
复杂度：$O(n^2)$

算法2-2

考虑如何优化。发现对于一个旅客 (s_i, t_i)，另一个旅客 (s_j, t_j) 和它不相交当且仅当$t_j \leq s_i$ 或 $s_j \geq t_i$。因为 $s_j < t_j$，于是就求一下关于 t_i 的前缀和和关于 s_i 的后缀和就行了。
复杂度：$O(n)$

算法3

考虑第二问，就是覆盖旅客数最多的区间覆盖的旅客数。这样用一个前缀和找一下覆盖最多的区间就行了。
复杂度：$O(n)$

结合算法3和算法2即可。
复杂度：$O(n)$
期望得分：100分

b.施工

算法1

对于每次询问把那条边删掉跑一遍最短路和原最短路比较一下就行了。

复杂度：$O(q*mlogn)$
期望得分：30分

算法2

考虑 S 到 T 的最短路图，即所有 $dist(S,u)+dist(v,T)+w$ 的边组成的有向图。这一定是一个有向无环图（DAG）。那么一个 S 到 T 的最短路是且仅是一条最短路图上 S 到 T 的路径。

- 如果一条边不在最短路图上，则这条边删掉一定不会对最短路产生任何影响。
- 如果一条边在最短路图上，则考虑这条边删掉之后这个最短路图还连不连通就行了。

于是变成了DAG上删掉一条边判 S 到 T 的连通性问题。有两种做法:

1. 如果一条边被删了就不连通，那么这条边一定是原图变成无向图之后的割边，跑一遍 $Tarjan$ 就行了。

2. 假设给每条边新建一个点 e' ，那么就加两条边 (u, e')，(e', v) 。那么一条边删了就不连通当且仅当 e' 是原图的 一个支配点。从 S 开始跑一遍 DAG 上的支配树就行了， 然后所有支配点就是 T 在支配树上和 S 路径上的点。 至于 DAG 上支配树怎么建， 就直接按拓扑序考虑， 然后每次考虑点 u 的时候，考虑所有 $(v, u) \leq E$ 的边， 把所有 v 求个 LCA 就行了。

复杂度： $O(n)$ 或 $(nlogn)$ 。
期望得分： 100分

c.组队

枚举中间的数，容易发现左边和右边的数一定是满足条件里正数最大的，或者负数最小的。我们就可以对正数和负 数分别维护一个 multiset, 查询使用 lower_bound 或者 upper_bound.

时间复杂度 $O(n \log n)$

d.距离

考虑计算出每个点为根时的距离和。挑出树上边权最小的边（记其边权为 w），将树按其分成两半，那么左半边的点到右半边的点，距离都是 w。 因此，可以先删去这条边，递归到左右两棵树算出答案；然后给左半边的每个点的答案加上 $w\times$右半边点数，右半边类似。

然后我们可以把这个过程倒过来做，就变成了按边权从大到小排序，依次加入边，合并两个连通块。同时维护连通块的大小和块内最大答案就可以了。使用路径压缩的并查集，时间复杂度 $O(n \log_2 n)$。