Problem 1 校门外的树（tree）

题目大意：要求给出一个 N 个数的数列，数列中每个数字的范围为 1 到 N，且数列要满足一系列诸如给定区间[Li,Ri]中没有重复数字的要求，问字典序最小的合法方案是什么。

解法：贪心。

首先，若有区间包含的情况，被包含的小区间显然可以忽略不做考虑。然后，考虑所有区间都不重叠的特殊情况，答案肯定是每个区间内数字依次为 1，2，3……区间外数字全是 1。

接着我们思考，若只有两个区间互相重叠了一小部分，最优答案是多少。例如，若两个区间[1,5]和[4,8]，那么，因为要求字典序最小，前 5 个数字肯定依次为 1，2，3，4，5，而注意选取第 6 个数字时，由于 6 仅在区间[4,8]中而不在区间[1,5]中，前 3 个位置上的数字 1，2，3 是可以“回收”重复利用的。

于是，这启发我们可以用一个集合保存着当前位置可以选哪些数字，从左往右每次选取数字时，都从集合中选出最小的数字，然后把这个数字从集合中删去，直到某个区间结束，再把刚结束的这个区间的前半部分（也即不与未结束的区间重合的部分）上的数字“回收”，重新加入集合中来。若某段区域不被任何区间覆盖，则特判一下一律填数字 1。

由于我们每次只需要从集合中选最小值并删掉，或者加入元素然后维护最小值，使用 priority_queue 或者 set 都可以。

Problem 2 放假回家（gohome）

题目大意：给定一个 N 个数的数列 a[i]和数字 M，对于每一个 i，计算出在 1~i-1 中，最多能选多少个数字，使得这些数字的和不超过 M-a[i]。

解法：权值线段树。

显然会选取 1~i-1 中数字最小的若干个数字，若对每个 i 都对 1~i-1 重新排序，或者使用插入排序后再统计答案，期望得分为 30 分。

但注意到第 4 第 5 个数据点 M 小于等于 100，所以我们使用插入排序时可以只维护前 100 个数字（若存在 a[i]=0 的特殊处理）的排序后的结果就好，期望得分 50 分。

满分做法可以使用权值线段树，线段树上每个结点维护对应权值区间范围内有多少个数字以及这些数字的和是多少，查询时每次判断左子树对应区间范围内有多少数字，相加后的和是否超过限制，超过则向左递归否则向有递归。时间复杂度 O(NlogN)，期望得分 100 分。

Problem 3 分糖果（candy）

题目大意：给定一个长为 N 的数列 a[i]，其中数列中数字有正有负。要求选出数列中前若干个数字，分成 K 段，使得每一段的和的最大值尽可能小。

解法：二分+DP+离散化+树状数组

要求最大的最小，第一时间想到二分。关键在于二分答案 mid 后怎么判断。

当 K=1 时，显然只需要看一下是否存在某个前缀和小于等于 mid 即可。期望得分 20 分。

当 K 不等于 1 时，我们可以使用 DP 来解决。用 f[i]表示前 i 个数字中，最多可以分成多少段，使得每一段的和都小于等于 mid。若存在 f[i]>=K，则 mid 是可行解，往小的范围继续二分，否则 mid 不合法，往大的范围继续二分。转移方程如下：

f[i] = max(f[j]+1), 其中 S[i]-S[j]<=mid

S 数组为数列 a 的前缀和数组，正确性显然。时间复杂度为 O(logRange*N^2)，结合 K=1 的情况，期望得分 50 分。

我们发现上面这个 DP 很慢，而且主要是状态转移太慢了，我们来思考如何优化状态转移。当 S[i]和 mid 固定时，符合条件的 j 就是那些满足 S[j]>=S[i]-mid 的 j，而转移时就是在这些符合条件的 j 里面找一个 f[j]的最大值。很容易想到使用数据结构来优化。

理论上我们可以使用权值线段树，但这里由于 a[i]的范围很大，其部分和 S[j]的范围就更大了，使用权值线段树容易超时。而且我们发现 S[j]是已知的且不变的，所以我们使用预处理离散化+树状数组会更好：先预处理出 S 数组，然后从大到小排序后离散化，建立树状数组记录对应前缀和大小为 S[j]时，最大的 f[j]是多少。那么接下来每次查询时就是查询树状数组中某个前缀的最大值。

时间复杂度为 O(logRange*NlogN)。期望得分 100 分。