本题中将所有比赛的结束时间排序，然后依次进行贪心：如果能够参加这场比赛，就报名参加；如果这场比赛和上一场冲突，就放弃。贪心本身的算法复杂度是 $O(n)$，但是排序的算法复杂度可达 $O(n\log n)$，所以时间复杂度的瓶颈在排序上。考虑值域范围不大，也可以考虑使用计数排序来优化。

12.2 哈夫曼编码

例 12-4 分卷子。某校要将一摞试卷按照等级分类。各个等级对应的成绩区间是：A(85, 100]，B(70, 85]，C[60, 70]，D[0, 60)。每次分卷子，只能将一摞卷子分为两堆，其中一堆包含了所有某些等级的卷子；另一堆包含所有另一些等级的卷子（换句话说，不会有两张相同等级的卷子同时出现在两边）。分好的卷子还能继续再分，直到分成 4 堆为止。已知各个等级的卷子的数量，请设计方案使分类比较次数总和最小。

分析： 比较次数的总和还能因为不同的分类方法而不一样？假设 A、B、C、D 分别有 10 人、13 人、14 人、5 人，然后观察图 12-5 所示的两个例子。

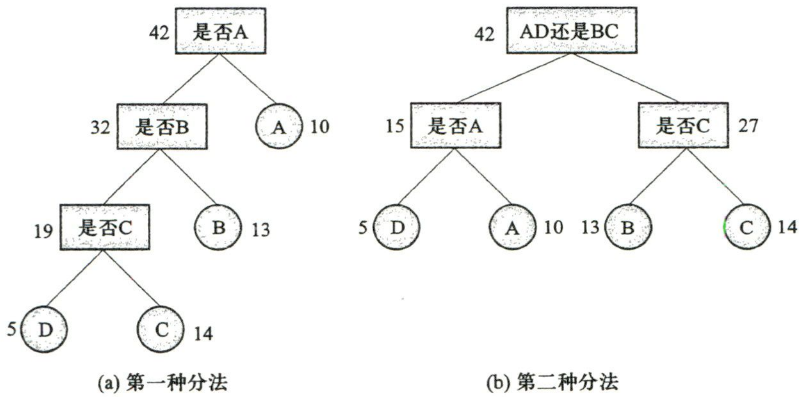

图 12-5 两种分卷方法

第一种分法中，A 分了 1 次，B 分了 2 次，C 和 D 各分了 3 次，一共分卷次数是 $10\times1+13\times2+14\times3+5\times3=93$，而第二种分法中，每种卷子都分了 2 次，一共分卷次数是 $2\times(5+10+13+14)=84$，显然第二种方式分卷次数要少一些。

在图中标上各个等级的人数和每次分卷时需要处理的试卷数，可以发现每次分卷次数刚好就是它分成两堆的分卷次数或者单等级的试卷数，比如对于第一种分法，判断是否 C 的次数等于 C 和 D 的试卷数之和，而判断是否 B 的次数刚好等于判断是否 C 的次数加上 B 的试卷数……直到最开始分类的时候，刚好就等于所有试卷数之和。因此可以做出假设：数量越多等级的卷子应该先分出来，数量较少的等级卷子可以多分几次在分出来，这样就可以减少分卷次数总和了。

可以大胆假设：从最开始分出最多的等级卷子，然后分出第二多的等级的卷子……直到全部分完这样的贪心策略。注意，如果此时能意识到必须先给出证明，那就说明前面的内容认真读过了。很可惜这样的分类方式并不是最优的，分卷总和还不如第二种分法。只要构造出一个反例就可以证明这是错误的贪心。