解题报告

7 月 20 日：排序，快速幂，暴力……

A: 二分法减少运算的时间成本

B: 快速幂减少运算的时间复杂度，第一次了解一个数的二进制形式的作用实例

C: 同样的快速幂运用+矩阵的运算，递推公式与矩阵的乘法运算的结合

D: 简单的排序问题。排序的多种方法，冒泡，快速，归并……

E: 排序问题，有优先级的考虑，也是这道题让我知道 pair 的数组，也可用 sort 进行排序

7 月 21 日：贪心，搜索……

A: STL 中 map 的运用

B: 我的方法是遍历数组，后一个与前一个相同的就加 1，不同就记录当前的计数，并将当前计数改为 1，直到末尾

C: 深度优先搜索？还是广度优先搜索？开一个二维数组记录走到当前位置的步数，用两个一维的数组来表示前进的方向

D: 用 string 来存放输入，标点是关键，分为标点前的一串数与标点后的一串数，再进行比较

E: 暴力!!! 找出 1 到 1000 所有数的因子个数，并依序存入数组中，对输入进行一一对比找出第一个匹配即可

F: 分阶段处理。首先对数据进行整理，化简并进行排序；然后找出一串可以同时进行的，若还有剩余，则将时间加 1，如此找下去，直到不再有剩余

G: 从小到大排序即可

H: 通过集合关系将做坐标面上的点与点之间的关系，转换成坐标轴上区域之间的关系，然后通过不停地截取公共区域与后一块区域进行比较，最后得出答案

I: 将每头猪分开看，计算出一千克的猪运到 i 村庄所花得费用，进行排序，将每头猪的费用进行排序，再利用排序不等式计算出最大收益

J: 真心佩服田忌能想出这么逆天的策略。1、比较速度最快的马，若能赢，则赢掉；若输，则用最慢的马比掉对方最快的马；若想等，则比较最慢的马。2、比较最慢的两匹马，若能赢，则赢掉；若不能赢，则用最慢的马比掉对方最快的马

7 月 22 日：动态规划问题

A: 第一题第一次写并没有用动态规划，自己写的程序怎么改都超时了，就去问了子赫大神，他也没有用动归，但是在遍历数组之前他维护了很多个值，一个记录当前的最小和，一个记录其坐标，一个记录其最大差值，及最大子数列，可以使得只需要遍历一遍数组就能找到答案。听完后我都觉得 66666666

B: 典型的动归问题，经典数塔。我采用了自上而下的递归形式，并维护一个二维数组，记录下当前位置的最大值。每个位置的最大值，就是自身加上其左下位置的最大值和右下位置的最大值中较大的一个

C: 当前位置的最大路径就是在周围的四个方向中找到能走并且路径最大的一个，再加上一。同样一维护一个二维数组记录每个位置的最大路径

D: 这是我唯一一道上网去看了代码的题，实在抽象不出来它的状态，而且就搞明白策略和网上的代码都花了我不下三个小时（智商太低没办法）。是用了一个三维数组记录当前位置的最大苹果数量，第一维坐标表示时间，第二维坐标表示可以移动的最大次数，第三维表示第几棵树……后面的问题就比较好推了，但是初始状况和无法达到的状态会导致出现很多细节上的问题

E: 若最后一个字符一样，则比较两个字符串的前 n-1 个字符；若最后一个字符不一样，则 1、比较第一个字符串的前 n-1 个字符和第二个字符串的 n 个字符；2、比较第一个字符串的 n 个字符和第二个字符串的前 n-1 个字符；取两种情况中所得子字符串最长的一个。我采用的是自下而上的住哪个台方程，比较容易实现

F: 维护一个一维数组 f[n]，其下标 i 表示以处在改下标 i 的值结尾的最大的和，其等于在前 i-1 个数中离该下标最近并且比该值小的数的 f 加上该值

7 月 24 日：

A: 哈夫曼树 = 最优二叉树，利用最小堆。首先将输入的数存入最小堆中，该数组中最小的数是该堆得根节点，其次小的的数处于根节点的左右节点中。每次取出最小的两个数，从该堆中删除，并将其和加入到该堆中，每次进行这样的一次操作都需要维护最小堆的性质（通过自己手写了一次堆，感觉对堆得认识透彻了不少）

B: 简单的排序问题，我用了 STL 中的优先队列

C: 运用散列表问题。第一次写的散列表，对其 key 值我需要在散列表中一一作比较找到其对应的 key 值，结果超时。然后重新写了一种散列表，其每个位置都有指针指向，这些指针又存于一个数组中,这样就支持直接通过 key 的下标索引，我觉得时间应该缩短了不少。但是还是超时，看了一天都没解决超时的问题，然后又是子赫大神，他对我说：把所有的 cin 改为 scanf 就 ac 了……然后我改了,然后就真的过了……我一定要搞清楚其性能问题!!!!! 我会常用 scanf 的

——李茂琦 7.26