第二届计算机学院“Debug 杯”程序设计竞赛简易题解

01 A+B

本题是给出两个集合求它们的集合并的大小。这道题定位是一道签到题，主要考察 for 循环/使用数组标记的方法/排序/stl 集合 set 的使用。

Solution 1：对第一个数组里的所有的数，用一个标记数组 vis，标记所有出现的数（vis[a[i]]=1），然后对于第二个数组里的所有数，如果没有出现过（!vis[a[i]），则计数 cnt++。最后答案就是 n+cnt。（n 指第一个数组的数的个数）

Solution 2：对于两个数组的数，一起读入进同一个数组，并进行排序。对排序后的数去重并计算个数即可。

Solution 3：使用 C++中的 stl 容器 set，将两个集合的元素都放入，最后输出该 set 的大小 size 即可。

知识点：排序，stl 容器 set，循环遍历

By caozy623

02 采木头

本题是给出若干时间点（HH:MM:SS），这些时间点会增加若干数量木头。现在给出一些询问，询问两个时间点之间增加的木头数量。

Solution ：因为询问数量较多，直接 for 循环回答询问的复杂度会很高。所以可以考虑先把所有时间点转化成一个整数，然后设置数组 S[i] 进行前缀和维护数量。这样的话对于时间点 L 到 R 之间的数量询问可以直接由 S[R]-S[L-1]给出。还有一个问题可以存在时间点 00:00:00，这个时间的话导致可能出现 L=0，从而 L-1 是负数（下标为负数）。这时候可以把所有时间点+1，然后查询的时候也相应地+1 进行回答即可。

知识点：前缀和，模拟

By caozy623

03 逛公园

本题是给出一张联通的无向图，并定义一个点对(u,v)之间的幸福度为,找出从 u 到 v 的一条简单路径,使路径上所有点权中最小值最大，该最大值即为点对(u,v)之间的幸福度。然后要求计算所有点对之间的幸福度之和。

Solution ：对于问题的第一部分，其实是一个瓶颈路的模型，可以转化为求最大/小生成树。这里要求最小值最大，所以只需先求最大生成树，则答案一定在这棵生成树上。对于第二部分，计算点对之间的幸福度之和，直接对于每条路径去计算肯定会时间复杂度很高，所以可以考虑贡献法。对于每条边，计算它在哪些路径中提供了贡献（被这些路径所包含）。可以发现，按照 Kruskal 算法做最大生成树的过程中，每新加入一条边，它作为答案的贡献，应该是当前将它加进来的时候这条边两边的 size 大小的乘积。（因为这条边是当前最小的，并且有这么多的路径经过了它，所以需要加进答案）故只要按最大生成树的边的加入顺序，再用带权并查集记录下各个连通块的大小，即可算出答案。

知识点：瓶颈路模型，最大生成树（Kruskal 算法），贡献法。

By caozy623

04 该怎么办呢

有两种牌，一种牌是亵渎，效果是：对所有随从造成 1 点伤害，如果有随从死亡，则再次施放该法术。另一种是地狱烈焰，效果是：对所有角色造成 3 点伤害。现在给出一些随从和他们的生命值，让你计算一下使用哪种牌能消灭的随从数量更多。
（出题目的时候忘记了术士在烧绳的时候说的是哪句话了随便拿了法师的语音当做题目了 233）

Solution 1：对于地狱烈焰的效果，直接计算生命值小于等于 3 的随从数量即可。对于亵渎的效果，对于随从按照生命值大小从小到大排序。然后如果第一小的数不是 1，那么亵渎一定打不死一个怪。如果存在生命值为 1 的怪，亵渎会打第一轮，那么其他所有随从的生命值都会减少 1，这时候其实只需要找原本的生命值为 2 的怪是否存在（现在被打了一轮是 1 了）。依次类推，我们可以发现，只要原数列从 1 开始等差为 1 的等差数列中的数都是可以打死的，但是如果只要出现某个数断档了，那么这时候亵渎就不能继续进行了。所以只需要排序后从小到大遍历，对于 a[i+1]-a[i]>1 的时候 break 即可。而答案就是 i。（标号为 i 以及之前的怪都会被消灭）

Solution 2：因为数据范围较小，对于亵渎的随从数量统计可以使用数组标记的方法把所有随从的生命值标记一下数量。然后从 1 到 1000（生命值上限），枚举每个生命值的随从是否存在，然后如果有就加上这个生命值的随从数量，否则跳出循环。

By caozy623

05 AI

题目：

在很长时间的深度学习之后，czy 和 yjh 终于制造出了终极 ai。终极 ai 可以回答一切问题(据说价值千亿)。
可是在答复了很多很多的问题之后，ai 累了。ai 决定不再回答问题。当然，除了那些问题中涉及到 czy 和 yjh 两个人。
也就是说，如果问题中涉及到了 yjh，输出 ai:yjhnb!。如果问题中涉及到了 czy，输出 ai:czynb!，如果不涉及两人，输出 ai:no response。

Solution：

对于一整行的字符串，使用 gets（c 语言）/getline（C++）/nextline（Java）读入即可。

对字符串进行判断，设置两个标记变量 f1,f2 分别表示是否有 czy 和 yjh。

如对于“czy”的判断对于字符串所有相邻的三位，是否满足 s[i]=='c' && s[i+1]=='z' && s[i+2]=='y'即可。

输出对应回答。

By nowhere

06 吃甘蔗

题目：

众所周知的是，jmatrix 同学非常喜欢吃甘蔗，所以他经常会去买甘蔗吃。

所以，他去买了一根 n 节的甘蔗。由于 jmatrix 一次吃能吃掉一节，所以他每次只会从甘蔗头或者甘蔗尾砍下一节然后吃掉。每节甘蔗有不同的甜度，另外 jmatrix 同学还有一个习惯，越晚吃，满足度系数越大。满足度等于每一节甘蔗的甜度乘上满足度系数的和。另外，为了简化题目的难度，贴心的 jmatrix 同学设置满足度系数为 1 到 n。即第一次的满足度系数为 1，第二次为 2，依次类推。

请输出 jmatrix 吃掉这根甘蔗后所能获得最大的满足度。

Solution：

区间 dp。从最后一次吃甘蔗的位置开始计算贡献，每次都能往左边或右边延长一个单位的距离，计算贡献，求最大值即可。

dp[i][i+l-1]=max(dp[i+1][i+l-1]+(n-l+1)×a[i],dp[i][i+l-2]+(n-l+1)×a[i+l-1])

复杂度 O(n^2)

By nowhere

07 能量不守恒定律

题目：

这个世界上有两种能量，a 能量和 b 能量。

小花作为一个魔法师，每天的工作就是根据现有的能量资源去产生新的能量资源。众所周知的是，由于能量不守恒定律，小花所拥有的能量会越来越多！

产生能量的规则是这样的：小花每一天只能释放两个魔法，第一个是元素克隆，可以将 b 能量克隆相同的数目，并转变为 a 能量。第二个是能量跃迁，会根据你现有的 a 能量和 b 能量的数目，产生 a 和 b 的最大公约数的能量(不消耗原来的能量)。

小话会严格按照魔法守则里的规定，即每天先做元素克隆，再做能量跃迁。

请问第 k 天之后，小花拥有 a，b 能量的数目分别是多少。

Solution：

本题需要会求最大公约数(gcd)的知识（辗转相除法）。同时需要注意答案可能超过 int，需要使用 long long 储存答案。其他部分按题意模拟即可。

By nowhere

08 图书管理员：

先对 a 数组做一个前缀和得到 sum[i](表示 [1,i] 层书架上的书籍数量总和)显然 sum[i]是有单调性的。于是我们便可以二分查找 sum 数组中第一个大于等于 b[i]的一个位置 pos，那么这本书应该被放在 pos-1 层，第（b[i]-sum[pos-1]）号。

时间复杂度 O(mlogn)

By lansebingmeng

09 逃离迷宫:

宽度优先搜索（BFS）模板题
但是注意到这里有多个起点，一个终点，如果我们对每一个起点做一下 BFS 显然会超时。
于是逆向思维，从终点 bfs 到第一个起点便输出答案。
PS：当然其实如果对于多个起点全部放入队列进行多源点 BFS 一样可以解决。当然如果对于多起点分别进行深度优先搜索（DFS）肯定会超时。

时间复杂度 O(n*m)

By lansebingmeng