

注意事项

1. 普通PC（包括我们的OJ服务器）每秒的基本运算次数 $X * 10^8$ ，在绝大多数情况下，估算运行时间 **只需要考虑数量级**，不考虑前面的常数。这一点常用于**估算程序运行时间和估算算法所需的复杂度**
2. 语言是整数是32位的，其取值范围为 $[-2147483648, 2147483647]$
3. 比较大的数组（如超 10^5 规模）要开在全局变量空间中
4. **碰到==WA (Wrong Answer) ==**可以自己尝试动手生成一些简单到复杂的样例，测试自己的程序。具体可以通过对stdin和stdout进行重定向实现，如 `freopen("test.in", "r", stdin);` 和 `freopen("test.out", "w", stdout);`

22-E1-1 缺少的数字

给定一个数组A，A的元素为 $[0, n]$ 中的n个数字，请找出在 $[0, n]$ 区间内，未在数组A中出现的数x。

思路

数组标记后判断，注意创建全局数组可避免爆栈

22-E1-2 相同的卡片

Alice和Bob分别拥有一些卡片，卡片以数字编号，Alice拥有的卡片以数组A表示，Bob拥有的卡片以数组B表示。现在给出数组A和B，你需要求得Alice和Bob共同拥有的卡片种类（以递增顺序），若无相同种类卡片则输出-1。

举例来说， $A=[1,2,2,3]$ ，表示Alice拥有1张卡片1，两张卡片2，1张卡片3。 $B=[2,2]$ ，表示Bob拥有两张卡片2。则他们共同拥有的卡片类型为卡片2，答案为2。

思路

因为存在递增，创建两个数组的指针（索引），指针递增判断即可。

22-E1-3 逆序对

输入一个长度为 N 的数组，你需要求出数组中逆序对的数量（答案对10,000,007取模）

逆序对即对于数组A，满足 $i < j, A_i > A_j$ 的 (i, j) 对

思路

在归并排序过程中记录逆序对数量，注意取模运算

22-E1-4 h指数

给你一个整数数组citations，其中citations[i]表示研究者的第i篇论文被引用的次数。计算并返回该研究者的h指数。

h指数的定义：h代表“高引用次数”，一名科研人员的 h指数是指他的n篇论文中至少有h篇论文被引用了至少 h次。

如果h有多种可能的值，h指数是其中最大的那个。

思路

首先，理解h的含义

将引用次数从小到大排序，之后从大到小遍历，h默认0，遍历过程中 $citation[i] > h$ 则 $h += 1$ ，最后输出h

22-E1-EX 次数排序

对于整数数组A，请按照每个数字出现次数升序排列后输出。

当出现次数相同时，按照数字大小升序排列

数据规模：

1. 数组元素 $-10^9 \leq \text{nums}[i] \leq 10^9$
2. 数组大小 $n \leq 1000000$

思路

数组元素较大，无法直接进行计数排序

通过创建包含（数值，出现次数）的结构体，重载compare()方法，即以“出现次数”作为结构体比较大小的依据

维护出结构体信息后，对结构体进行排序，输出即可。

22-E2-1 数据库查询

勤奋的小明为了预习下学期的数据库课程，决定亲自实现一个简单的数据库系统。该数据库系统需要处理用户的数据库插入和查询语句，并输出相应的输出。具体来说，用户的输入共包含若干条插入语句和查询语句。其中每条插入语句包含一个主键（唯一标识） id 和一个字符串 s_{id} ，表示在数据库中插入一个主键为 id ，属性值为 s_{id} 的条目。每条查询语句包含一个字符串 k ，表示在数据库总查询是否有条目属性值为 k ，若存在属性值为 k 的条目，则输出该条目对应的主键（输入保证至多有一个条目满足要求），否则输出一个-1表示不存在这样的条目。

思路

建立值->键的映射关系

1. 字符串哈希
2. RB Tree
3. 比较偷懒的写法（不推荐）：STL库中的 `map<string, int>`

22-E2-2 树的遍历

给定某二叉树的前序序列和中序序列，输出该二叉树的后序序列。（输入的前序遍历和中序遍历的结果中都不含重复的数字）

思路

通过递归来做，把握前序的第一个一定是当前树/子树的根节点，利用这一信息在中序序列中找出左子树和右子树之后进行递归

22-E2-3 内存分配

C语言中需要申请一块连续的内存时需要使用malloc等函数。如果分配成功则返回指向被分配内存的指针(此存储区中的初始值不确定)，否则返回空指针NULL。

现在小明决定实现一个类似malloc的内存分配系统，具体来说，他需要连续处理若干申请内存的请求，这个请求用一个闭区间 $[a_i..b_i]$ 来表示。当这个区间和当前已被申请的内存产生重叠时，则返回内存分配失败的信息。否则返回内存分配成功，并将该区间标记为已被占用。

假设初始状态下内存均未被占用，管理的内存地址范围为 0_{1GB} (02^{30})。

思路

需要频繁对整段区间进行修改（0->1）、查询操作，不能直接在数组上进行操作。
可以使用红黑树 / 树状数组 / 线段树区间操作

22-E2-4 危险品放置

现有若干危险品需要放置在 A, B 两个仓库。

当两种特定的危险品放置在相同地点时即可能产生危险。我们用危险系数 $\alpha_{i,j}$ 表示危险品 i, j 放置在一起的危险程度。一些危险品即使放置在一起也不会产生任何危险，此时 $\alpha_{i,j} = 0$ ，还有一些危险品即使单独放置也会产生危险，此时 $\alpha_{i,i} > 0$ 。

定义两个仓库整体的危险系数为 $\max(\max_{i,j \in A} \alpha_{i,j}, \max_{i,j \in B} \alpha_{i,j})$ ，即放置在一起的所有危险品两两组合的危险系数的最大值。

现在对于一组给定的危险系数，需要设计方案使得整体危险系数最小。

思路

并查集 + 贪心，贪心将危险程度较大的物品放置到两个不同的仓库中，利用并查集维护冲突关系，直至不能无法放至两个仓库时得到最小的危险系数。

22-E3-1 最大和

对于一个长度为n的整数数组，请你找出一个具有最大和的连续子数组，输出其最大和。

思路

动态规划， $dp[i]$ 表示以 i 结尾的连续子数组最大和，通过 $dp[i] = \max(num[i], dp[i-1]+num[i])$ 来更新
分治，递归

扫描一遍，在每次元素累加和小于0时，从下一个元素重新开始累加

22-E3-2 最小路径

对于一个元素均为正整数，大小为 $n*m$ 的网格，Alice从左上角(1,1)出发，每次只能选择向右或向下走一格，最终到达网格右下角停止。

到达网格右下角存在若干路径，请你找出路径上各点的数字总和最小的路径，输出其数字总和。

思路

动态规划，根据可走的方向维护即可， $dp[i][j]$ 记得考虑 i,j 在边界的情况

22-E3-3 多重背包

现有一个背包可以容纳总重量为 W 的物品，有 n 种物品可以放入背包，其中每种物品单个重量为 w_i ，价值为 v_i ，可选数量为 num_i 。

输出可以放入背包的物品的最大总价值。

思路

在0-1背包的基础上，在状态转移部分，根据当前物品可选数量进行更新（增加一轮循环）

22-E3-EX 正方形计数

现有一个 $n \times m$ 的矩形区域，其中每个单位区域可能有损坏。

要求找到地面上所有不包含损坏区域的正方形的个数。

思路

动态规划， $dp[i][j]$ 表示以 (i,j) 为右下角的正方形有几个

若 (i,j) 位置为1，则 $dp[i][j] = 0$;

若 (i,j) 位置为0，则状态转移 $dp[i][j] = \min(dp[i-1][j], dp[i][j-1], dp[i-1][j-1]) + 1$;

最后把dp数组求个sum就可以

22-E4-1 道路规划

沙漠上，新建了 N 座城市，用 $1, 2, 3 \dots N$ 表示，城市与城市之间还没有道路，现在需要建设这些城市的道路网络，需要在城市间修建道路。施工队给出了 M 条道路的预计费用信息，每条道路的预计费用信息可以表示为 $U_i V_i W_i$ （即如果要在 U_i 和 V_i 之间修建道路，预计费用为 W_i ），道路是双向的。现从 M 条道路中选择一定数量的道路来修建，使得这些城市之间两两之间可达（可以通过其他城市间接到达），你要求出达成上述条件的最少预算

思路

最小生成树问题，可以使用kruskal算法

22-E4-2 逃离迷宫

小明被困在一个迷宫之中，迷宫中共有 N 个点，标号分别为 $1, 2, 3 \dots N$ ，且迷宫只有一个出口。 N 个点之间互相有一些道路连通(单向)，两个点之间可能有多条道路连通，但是并不存在一条两端都是同一个点的道路。小明希望知道从当前位置 S 去往出口 T 的最短距离是多少。如果不存在去往出口的道路，输出 -1

思路

单源最短路径问题，可以使用dijkstra、Bellman-Ford算法

22-E4-3 货物运输

在一个工厂货物运输系统中共有 N 个节点, 编号为 $1, 2, 3 \dots N$, 节点之间有传送带(单向)连接, 每个传送带都有使用寿命, 传送带的寿命为一个数字 L , 表示在传送完 L 个单位的货物后, 传送带就会破损无法使用。现在需要从节点 S 向节点 T 传送货物, 求在当前传输系统中, 最多可以顺利传输多少单位的货物从节点 S 到节点 T

思路

最大流问题, 可以使用Ford-Fulkerson算法

22-E4-4 重复的子串

给定一个字符串 s , 请判断是否可以通过由它的一个子串重复多次构成。若能, 则输出该子串(最短的), 否则输出-1。

思路

KMP算法拓展, 在由重复子串组成的字符串中, 最长相等想后缀, 不包含的子串就是最小重复子串

利用KMP算法求得next数组, 接着通过next数组最后一个元素的指向的最长重复前缀位置来进行判断是否当前字符串是否为重复的子字符串

22-E4-5 字符串匹配

你需要求出模式串 P 在原串 S 中出现的次数, 模式串和原串中的字母均为小写, 数据有多组, 每组输出一个答案

思路

KMP算法拓展

22-E4-6 重新排列

对一个由正整数组成的数组 $arr[]$ 进行重新排列, 使得排列后的数组从前到后组成一个最大的数(数组内元素不可拆分)。

例如3 92 10 4数组重新排列后可得到最大的数字为924310。

输出结果可能非常大, 请以字符串输出。

思路

对于 A 和 B 两个数, 若 $A\&B$ (其中 $\&$ 表示连接) $> B\&A$, 则 A 应该排在 B 前面, 因为 A 为前的收益更大那么重载一个 $compare(A,B)$ 函数, 对所有数字依据“ $A\&B$ 是否大于 $B\&A$ ”进行排序, 最后最前往后输出即为答案。