Solutions for Pacific Northwest Region Programming Contest Division 2

中国人民大学 ACM 队

2020 年 2 月 21 日

目录

1 Alphabet

1.1 Description

我们定义一个小写字符串是"按字母表的",当且仅当它删除掉一些字符后,可以变为"abcdefghijklmnopqrstuvwxyz" 的字母表。

给定一个长度为 n 的小写字母字符串，询问至少插入多少个字符才能使其变成 "按字母表的"。

$n \le 50$。

1.2 Solution

不难看出后来添加的字母一定不会在变成 "abcdefghijklmnopqrstuvwxyz" (以下简写为 $a \sim z$) 被删去，不然没有加入的必要。

所以为了将原字符串变成 a ∼ z，我们考虑先删除再添加。

不难看出删除后的字符串字典序一定严格递增，不然不能仅通过添加得到 a ∼ z。

而严格递增的序列一定是 a ∼ z 的子序列，可以经过添加变成 a ∼ z。

所以任务变成寻找输入串的最长上升子序列，答案就是 (a ∼ z − 最长上升子序列长度)。

设以第 i 位为结尾的最长上升子序列长度为 f_i，得到转移方程：

$$f_i = \max\{f_j + 1\} \ (i > j \text{ 且第 } i \text{ 位字典序大于第 } j \text{ 位})$$

2 Barbells

2.1 Description

给定 n 个空杆，m 个圆盘，当杆左右重量相等时对答案有贡献，求出所有可能重量。

重量 $w =$ 杆重 $b_i + \sum$ 圆盘重 p_i

$1 \leq n, m \leq 14$

$1 \leq b_i, p_i \leq 10^8$

2.2 Solution

m 和 n 范围很小，可以接受指数级别算法，直接暴力枚举即可。

考虑三进制，某一位为 0 即把这个圆盘放到左边，为 1 就放到右边，为 2 则不选。

最后统计即可。因为答案需要从小到大输出且相同重量只计一次，所以可以用set统计答案。

3 Buggy Robot

3.1 Description

一个机器人要走出一个 $n \times m$ 的矩形网格迷宫，迷宫中的每个格子为障碍物或者为空。已知迷宫的布局、机器人的起始位置和迷宫的唯一出口位置。保证存在从起始位置到出口的路径。

已知一串给机器人的指令，指令由若干 U 、D 、L 或 R 组成，分别表示上、下、左、右四个方向。机器人会按照指令的顺序依次向对应方向移动一格，若碰到障碍物或迷宫边界则在原地不动。一旦机器人到达终点，后续指令都将被忽略。

给出的这串指令并不一定能使机器人到达终点，现在要求修改该指令使得机器人能走出迷宫，修改方法是每次可以花费单位代价来删除某一条指令、或者在指令串的任意位置插入一条指令。求最少花费多少代价可以使机器人按照指令能够到达迷宫出口。

$n, m \leq 50$，给出的指令串长度范围为 $l \in [1, 50]$

3.2 Solution

可以发现，机器人的状态由其所在位置和已经执行了多少条原指令串中的指令（包括删除的指令）所确定，这样的状态数为 $O(nml)$ 。而状态的转移只需枚举不进行指令的修改、删除下一条指令、插入一条指令这些决策即可。

将状态视作图上的节点，转移视作图上的边，则问题转化为了最短路问题。而图中的边权只可能为 0 或 1 ，因此只需对 bfs 算法稍加修改即可求出最短路。具体方法是使用一个双端队列，搜索过程中将通过边权为 0 的边扩展的状态加入到队首，通过边权为 1 的边扩展的状态加入到队尾。需要注意的是，这样修改之后一个元素可能在队列中出现两次，需要进行判断。具体实现时可以使用 C++ 标准库中的 std::deque 。

总时间复杂度为 $O(nml)$ 。

4 Cameras

4.1 Description

有 n 个房子，从 1 到 n 标号一字排开。每个房子可以选择装或不装摄像头，要求任意连续 r 个房子中至少有 2 个房子有摄像头。现在有 k 间房子已经安装了摄像头，问最少还需要安装几个摄像头。

$2 \le n \le 10^5, 0 \le k \le n, 2 \le r \le n$。

4.2 Solution

不难想出如下的贪心策略：从左到右扫描每个区间，如果该区间内不足 2 个摄像头，则从右边未安装摄像头的房子开始安装起。这是因为越靠右边的摄像头越能被后面更多的区间共有。如果从右向左扫描区间，则优先安装靠左的摄像头。

下面证明该贪心策略的正确性：

1）最优子结构

设一个最优解 S 有一个子问题 S' 表示 $1 \sim t$ 房子的摄像头安装情况。如果 S' 不是最优的，则存在一个比 S'' 更优的解，使得 S 更优。此时，S 就不是最优解，与假设不符。因此，若 S 最优，其子问题 S' 必然最优。最优子结构得证。

2）贪心选择性质

如果一个区间不足两个摄像头时，不选择从最右（不妨记为 y）开始安装，并安装在 x 处。则右边的所有区间不会有更多的摄像头，其中某些区间 $[x, x+r-1]$ 到 $[y, y+r-1]$ 还会减少摄像头，显然使解更劣。贪心选择性质得证。

5 Contest Score

5.1 Description

有 n 个题，解决它们所需要的时间分别为 $a_1, a_2, \ldots, a_n$，执行以下步骤：

1. 取出前 k 个题。
2. 选出这 k 个题中用时最小的一道题，解这道题并花费当前时间 + 解题所需时间的代价。
3. 将下一道题加入这 k 个题中。
4. 如果有未解决的题，回到步骤 2。

问最后的代价之和。

$1 \leq k \leq n \leq 300$

$1 \leq a_i \leq 10^6$

5.2 Solution

使用小根堆，先用前 k 个数建堆，每次取出最小值并计算代价，并将下一个数放入堆中（如果存在未入堆的数），直到所有的题都被解决。

6 Equality

6.1 Description

给出一个形如 $a + b = c$ 的等式，满足 a、b、c 均为一位正整数，且该等式包含空格在内共有 9 个字符，如果 a 与 b 的和等于 c，则输出YES，否则输出NO。

6.2 Solution

考虑直接按照题意模拟，读入 a、b、c 三个数字，并判断 $a + b$ 是否等于 c，并在相等时输出YES，不相等时输出NO，即可通过此题。

7 Gravity

7.1 Description

给定一个 $n \times m$ 的二维平面图，每个格子可能是苹果'o'，障碍'#' 或者空地'.'

受到重力作用，苹果会自上而下地下落直到触底或者碰到障碍和苹果，会停在底边或者障碍和苹果上方的格子处。

求最终这个二维图的样子。

$n, m \leq 50$

7.2 Solution

不难看出列与列之间的苹果是相互独立，不受彼此影响的。

所以我们对于一列单独考虑。

在同一列中的苹果会产生影响，但是苹果的相对位置不会发生变化，所以我们先处理低处苹果，因为它的最终位置不会受高处苹果最终位置的影响。

然后按照这个顺序模拟自由下落的过程，一直下落直到碰到物体或底边停下即可。

8 Islands

8.1 Description

给一个 $N \times M$ 的矩阵，矩阵由 L,W,C 组成

其中 L 表示岛屿，W 表示海洋，C 表示云朵

云朵下可能是岛屿或者海洋

本题要求确定一种方案，使得连接的岛屿数量最少。两个岛屿是否连接，当且仅当两个岛屿之间在四方向（上下左右）相邻。只需要输出最少岛屿数量。

$1 \le N, M \le 50$

8.2 Solution

考虑一个连接的岛屿向外拓展，如果和该岛屿直接连接的为 C，将其改为 L。此方法并不会增加已连接岛屿数量。而如果直接连接的为 L，表示两个原本不连接的岛屿相互连接，会使岛屿数量 -1。迭代多次后可以发现，两个岛屿无法连接当且仅当两个岛屿被 W 阻隔。

因此有如下算法，遍历每一个未被标记的 L，对 L 进行四方向BFS。如果遍历到 L 或者 C 则继续（并标记），如果遍历到 W 或者边界则停止。BFS结束后答案计数 $+1$。最后输出的答案为最少岛屿数量。

9 Mismatched Socks

9.1 Description

你现在有 n 种颜色的袜子，第 i 种有 k_i 只，问最多可以匹配多少双颜色不同的袜子。

9.2 Solution

事实上, 如果袜子最多的那一种颜色的袜子数量没有超过总数量 s 的一半, 那么一定可以两两配对到只剩下 $s \mod 2$ 只袜子。我们可以把袜子按颜色顺序排成一列, 依次编号为 $1, 2, 3, \ldots, s$ 。然后将 1 与 $\frac{s}{2}+1$, 2 与 $\frac{s}{2}+2 \ldots \frac{s}{2}$ 与 s 配对，一共 $\frac{s}{2}$ 对。

如果颜色最多的袜子超过了总数量的一半，不妨设其有 m 只。则只能将剩下所有的袜子和他配对，共可配对 $s-m$ 对。

于是最终的答案就是 $\min(\frac{s}{2}, s-m)$ 。

10 Postman

10.1 Description

有一名邮递员要在一个坐标轴上从邮局送信到 n 个地点，每个地点有 m_i 封信件。

邮局在位置 $X = 0$ 处，邮递员每移动 1 个单位长度需要 1 个单位时间。邮递员一次至多带 k 封信。求邮递员从出发到送完信返回邮局至多需要的时间。

$n \le 1000, k \le 10^7$,

$|x_i|, m_i \le 10^7$。

10.2 Solution

当邮递员到某一半轴送信时，若要去另一半轴则必须返回邮局，因而将两个半轴分开考虑。

单次送信时间只由信件中需要送达的最远距离决定，因而设计贪心策略：每次从远至近取 k 封信进行寄送，裸算法复杂度 $\mathrm{O}(\frac{nm}{k})$，注意到可能超时，优化后变为 $O(n)$。

11 Six Sides

11.1 Description

给出两个六面的骰子，每面上的值在 $1-6$ 之间，且每面向上的概率相等。现在不断投掷这两个骰子，直到两个骰子向上面的值不等，并判定值大的骰子获胜。

问第一个骰子获胜的概率是多少?

11.2 Solution

令在一轮投掷中，第一个骰子获胜的概率为 P_{win}，平局的概率为 P_{draw}，输的概率为 P_{lose}。

则第一个骰子获胜的概率为

$P = P_{win} + P_{draw} \times P_{win} + P_{draw}^2 \times P_{win} + \cdots + P_{draw}^n \times P_{win}$
$= \frac{P_{win} \times (1-P_{draw}^n)}{1-P_{draw}}$

故 $ans = \lim\limits_{n\to\infty} P = \frac{P_{win}}{1-P_{draw}} = \frac{P_{win}}{P_{win}+P_{lose}}$

12 Three Square

12.1 Description

你被给予了三个矩形的长与宽，问其是否能在不重叠的前提下拼成一个正方形。

矩形的边长为不超过 100 的正整数。

12.2 Solution

签到题，矩形的排列只有两种方式：

AAA

BBB

CCC

或者，

AAA

BCC

BCC

我们只需要枚举这三个矩形的位置与边长的相等关系，大力判断即可。

13 Zigzag

13.1 Description

定义如果一个序列相邻元素在严格递增和严格递减之间交替，则称该序列是“之字形”的。其中第一对数字可以是严格递增的或严格递减的。对于给定序列，求其最长之字形子序列的长度。

13.2 Solution

与最长上升子序列做法类似，定义 $f_{i,0/1}$ 表示以第 i 个元素结尾的之字形子序列长度最大值，第二维表示该子序列最后一对数字是严格递增的/严格递减的。初始值为 1。

对于第 i 个元素和第 j 个元素 $(i < j)$，如果 i 元素 <j 元素，则 $f_{j,0} = \max(f_{j,0}, f_{i,1} + 1)$。

如果 i 元素 $>$ j 元素，则 $f_{j,1} = \max(f_{j,1}, f_{i,0} + 1)$。

f 数组的最大值即为答案，时间复杂度 $O(n^2)$。