

2019校招编程Solution

一、牛牛找工作

不同时间复杂度的Solution:

1.暴力搜索：（时间复杂度是 $O(mn)$ ）

使用最直接的排序方式进行挑选（主要实现了自己的二维数组排序）但是存在时间复杂度太高的问题。

2.优化了时间复杂度（ $O(n\log n)$ ）

报酬总是基于之前的最大值进行比较

二、被3整除

用了一种很奇怪的方法：按理需要将每一位的数字加上去然后看是否能给3整除，但是这样的话就会复杂度太高，所以用了一种很奇怪的想法，就是 $1+2+3+\dots+12+13$ 这样的方式来判断他是否被3整除。

三、安置路灯

只有两种情况需要判断，首先从第二个点开始判断，如果前面是路灯，那么就要在此处安置路灯，然后把考虑的点往后移两位，至于后移两位还有一种情况就是越界，这种情况看是否最后一位是灯那么还要安置一个灯。

四、迷路的牛牛

建立一个方向数组，使用循环数组，往左就减一，往右就加一。

五、数对

还是时间复杂度的问题，需要考虑从K开始，然后对于每一个考虑的数他对应的余数是成循环的，只需找出循环次数乘以符合条件的数量，再加上最后余下的数符合条件的个数。

六、矩形重叠

考虑每个矩形的左下角和右上角离散化（进行排序并剔除相同项），然后看成 $2N \times 2N$ 的方格，方格中的每个点的权值就是该点出现矩形的个数，扫一遍看最大值就找到。

七、牛牛的闹钟

对闹钟进行排序（由小到大），然后从最后开始搜索，加上用时小于等于到达时间就可以输出。

八、牛牛的背包问题

使用背包的原理，每一个物品可以选择放进去还是不放进去，state描述了这个状态，而这个状态可以分解为（1）第i个不放进去，前i-1个总体积小于等于W，

（2）第i个放入情况下，前i-1个总体积小于等于 $W - v[i]$ ，也就是说我们可以写出这样的状态转移方程： $state(i, w) = state(i - 1, w) + state(i - 1, w - v[i])$ ，这个公式的边界条件是：当 $w < 0$ 时， $state = 0$ ；当 $i = 1$ 时，而 $w > 0$ 且 $v[1] \leq W$ ，，状态有两种， $state = 2$ ；当 $i = 1$ 时，而 $w > 0$ ，且 $v[1] > W$ ，只能不放入一种， $state = 1$ ；例如：零食有1, 2, 4, $W = 10$ ，所以：

$$\begin{aligned} state(3, 10) &= state(2, 10) + state(2, 6) \\ &= state(1, 10) + state(1, 8) + state(1, 6) + state(1, 4) \\ &= 2 + 2 + 2 + 2 = 8 \end{aligned}$$

但是这道题这样考虑复杂度会比较高，只能AC80%，所以考虑总体积小于背包体积直接可以给出 2^n 的装法。同时应该注意背包体积用long long型。

九、俄罗斯方块

使用单维数组进行存储每一列的方块数，然后每一次更新之后判断所有列的方块数是否都大于当前的得分，如果是则得分加一。

十、瞌睡

这道题如果使用暴力搜的话时间复杂度是不能满足要求的，因此需要在输入的时候分别统计状态是一时的总的固定收益，同时用一个累加数组统计状态是0时的累加收益，然后判断累加数组相减后的最大值。如下：

输入: 6 3
1 3 5 2 5 4
1 1 0 1 0 0

醒着时的分值: 1 3 0 2 0 0
此分值固定为: 6

+

瞌睡时分值数组: 0 0 5 0 5 4 ^{可叫醒点}
瞌睡分值累加数组 arr: 0 0 5 5 10 14
i点叫醒所得额外分值: $arr[i+k-1]-arr[i-1]$, 不考虑数组越界时

part1

part2

十一、丰收

这道题就是二分查找，在查之前对输入进行一个累加数组的构造就可以了。