

51. N皇后 (与数独, 全排列问题类似): 是全排列问题的变种.

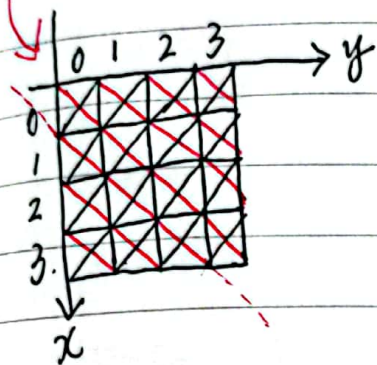
利用dfs解决.

皇后可以横着, 竖着或斜着沿对角线走.

不能相互攻击指: 没有两个皇后在同一行, 同一列或同一对角线上.

则由: n 个皇后放在 $n \times n$ 棋盘上可知性质: ① 每行只有一个皇后 ② 每列只有一个皇后.

③ 对角线上不能有多个皇后.



$$k = y - x + n$$

$$\begin{cases} y = x + k \\ y = -x + k \end{cases} \Rightarrow$$

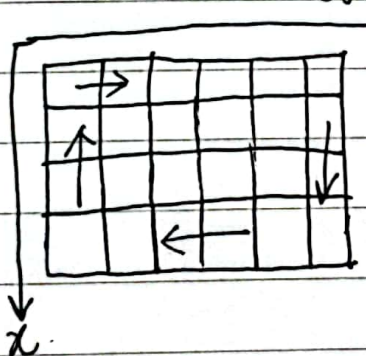
$k = y - x + n$. (红线需调整 $+n$ 以使得 k 为正数)
 $k = y + x$. (黑色对角线)

52. N皇后II: 与51题思路完全相同, 不过改成求数量.

53. 最大子数组和: (动态规划): 考查以 $nums[i]$ 结尾的子数组的最大值和.

方法二: 分治法. 见代码. 该方法可求出各区间的最长子数组和. (大规模查询时有优势)

54. 螺旋矩阵:



$\rightarrow (0, 1)$ 不断这样走, 直到越界
 $\downarrow (1, 0)$ 越界或格子重复走过时
 $\leftarrow (0, -1)$ 应换方向. 直到越界
 $\uparrow (-1, 0)$ 向也不能走为止.

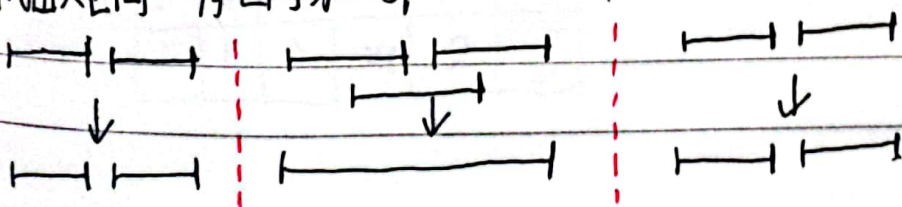
55. 跳跃游戏: 与45题思想一致.

56. 合并区间: 步骤: ① 按左端点排序 ② 若有交集, 更新右端点. 若没有交集, 保留当前
 时间: $O(n \log n)$

$r < a[i][0]$

$r \geq a[i][0]$

57. 插入区间: 将区间分三部分处理. 举例如下:



模拟题 $O(n)$

58. 最后一个单词的长度: 双指针算法. 倒着遍历 (双指针算法) 即可.

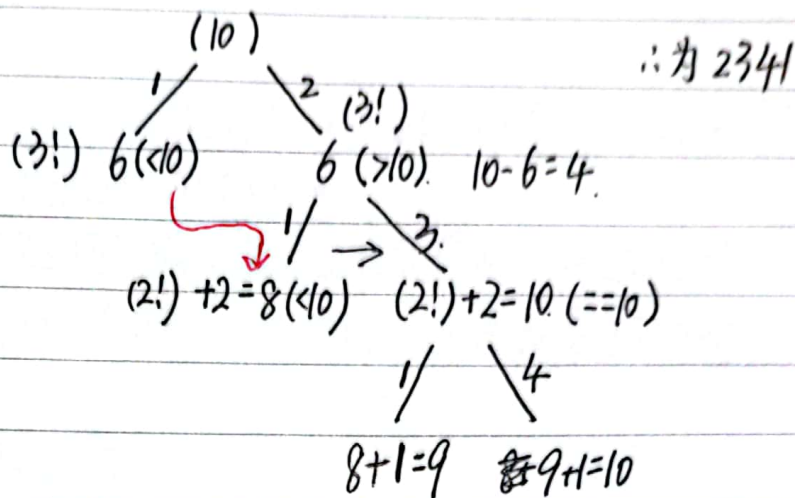
59. 螺旋矩阵 II: 与54题思路完全一样.

60. 第k个排列: 计数问题.

举例: $n=4, k=10$ 时.

一位一位的看.

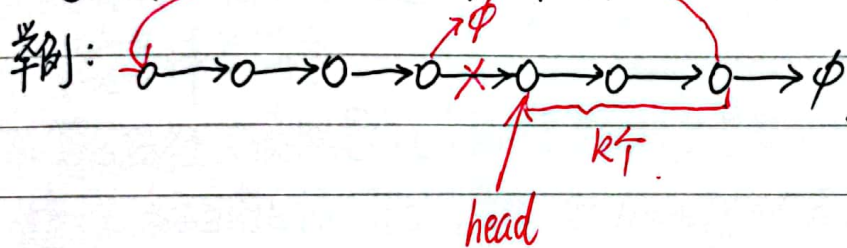
时间复杂度: $O(n^2)$



61. 旋转链表: 设链表中有几个节点.

① 若 $k \geq n$, 则相当于循环移动, 则应: $k \% n$.

② 观察: 相当于将链表的后 k 个节点移动到前面.



62. 不同路径: dp问题.

举例: 3×5 的格子

1	1	1	1	1
1	2	3	4	5
1	3	6	10	15

方法二: 组合思想: 由 $(0,0) \rightarrow (n-1, m-1)$.

需向右走 $n-1$ 次, 向下

走 $m-1$ 次. 共走 $n-1+m-1$ 次.

∴ 答案为: C_{n+m-1}^{m-1}

动规

令 $f(i,j)$ 表示从 $(0,0)$ 走到 (i,j) 有几种方案.

注意代码写法是比较巧的.

$$f(i,j) = \begin{cases} 1 & (i=0 \text{ 或 } j=0) \\ f(i-1,j) + f(i,j-1) & (\text{其他}) \end{cases}$$

63. 不同路径 II: dp问题. 与62题本质相同

举例:

1	1	1	1	X0
1	X0	1	2	2
1	1	2	X0	2

动规.

64. ~~最小~~ 最小路径和: 和 62.63 思路相同, 仍是 dp 法, dp 方法也相同.

65. 有效数字: 模拟题, 具体参考代码.

66. 加一: 高精度加法.

67. 二进制求和: 二进制高精度加法.

68. 文本左右对齐: 规则: ① 最后一行只需左对齐, 两个单词间只有一个空格, 无额外空格.

② 若一行只能有一个单词, 则该单词左对齐即可.

③ 一般情况: 左右对齐, 可在单词间插入额外空格使得单词分布的更均匀, 空格应尽可能分布均匀, 若无法分布均匀, 则左侧放置的空格数要多于右侧.

----- 空格

类似于双指针的程序, 详见代码.

69. x 的平方根: 二分问题. 目标: $ans^2 \leq x$ 中 ans 的最大值.

注意如何设置 l, r 及如何防止越界.

70. 爬楼梯: 简单 dp 问题. $f(i) = f(i-1) + f(i-2)$

注意怎样写简单.

（它和斐波那契数列类似 栈 的思路): $O(n)$ 算法.