71. 简化路径（模拟题）（实现时是类似栈的思路）：O(n)算法.

注意：1. 表示当前目标，即没有变化，如 /home/you/. 表示：/home/you

1.. 表示上层目录，如：/home/you/../ 即 /home/you/../abc/ 即：/home/abc.

↓ 代码中的处理单元.

72. 编辑距离（dp问题） O(n²)的算法（∵有n²个状态）

① 状态表示：f(i,j) 表示将a字符串 word1[1~i] 变成字符串 word2[1~j] 的所有操作中，操作步骤的最小值.

所需

② 状态计算：$f(i,j)=\min\begin{cases} f(i-1,j)+1 & (A[i-1]\text{变为}B[j],\ A\text{中}\ \text{删除}\ A[i]) \\ f(i,j-1)+1 & (A[i]\text{变为}B[j-1],\ A\text{中插入}\ B[j]) \\ f(i-1,j-1)+1\text{或}0 & (A[i-1]\text{变为}B[j-1].\ \text{若}A[i]==B[j],\text{加}0,\ \text{若}A[i]!=B[j],\text{将}A[i]\text{变为}B[j]) \end{cases}$

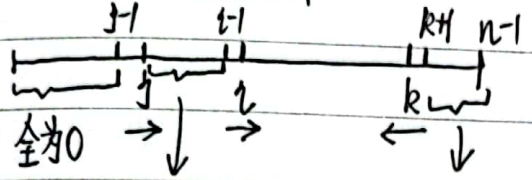
73. 矩阵置零（难点：使用原地算法）
思路：利用原数组记录状态。 } 用原数组的第一行表示每一列有没有0，用原数组的第一列表示每一行有没有0。

此外，用两个变量：r_0表示第一行有没有0，c_0表示第一列有没有0。

74. 搜索二维矩阵。（二分问题，将二维二分可转化为一维二分问题）

75. 颜色分类（本质上是双指针算法）（三指针）

图解：

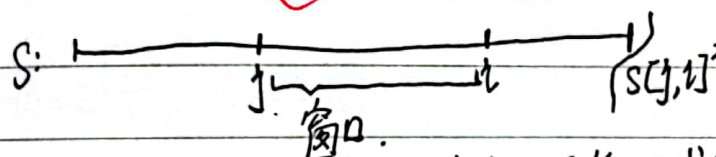


[0,j-1]：全为0　　[j,i-1]：全为1　　[k+1,n-1]：全为2。

考查a_i：
① $a_i==0$，swap(a_i,a_j)，j++，i++；
② $a_i==2$，swap(a_i,a_k)，k--。
③ $a_i==1$，i++
→ 可见每次迭代必有i++或k--

☆注意终止条件是i==k

76. 最小覆盖子串。（滑动窗口算法，双指针算法的应用）：O(n)算法



表示以字符S_i结尾的最小覆盖子串

性质：当i向后移动时，j一定也向后移动或不动，符合利用滑动窗口算法的特性。

详细实现见代码。

77. 组合：dfs算法，防止重复选取的方法就是必须按一定从前到后的顺序选取方案（以前讲过）

78. 子集：可以像77题一样用dfs递归写，不过本题有另一种迭代写法。

用二进制方式枚举子集。举例：数字1~3，集合{1,2,3}的子集。

二进制：	子集。
00	{3}
01	{1,3}
10	{2,3}
11	{1,2,3}

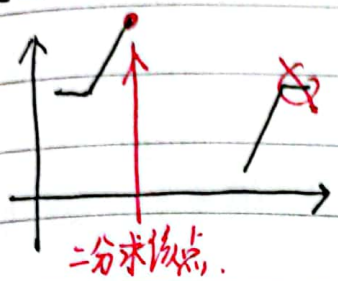
☆注意代码写法。

79. 单词搜索。dfs搜索问题。注意dfs中应用剪枝及时返回。

80. 删除有序数组中的重复项II：双指针算法 详见代码.

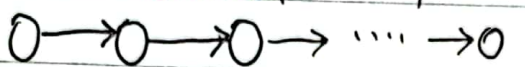
81. 搜索旋转排序数组II：注意本题用二分最坏时间复杂度也为O(n)，不如线性扫描. 直接.

不过还是写一下二分. 思路：删除末尾与开头的重复元素后二分.



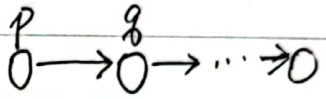
82. 删除链表中的重复元素II：双指针算法. 详见代码.

排序



(dummy)p p->next q=p->next->next.

83. 删除排序链表中的重复元素：仍是双指针算法.

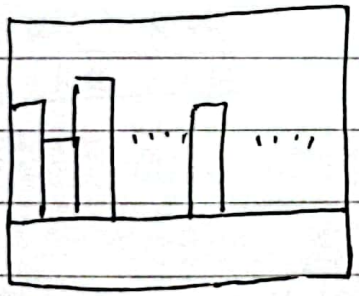


84. 柱状图中最大的矩形：单调栈的应用.

本质：找到某数左边第一个比它小及右边第一个比它小的数的位置 } 详见代码.

两个单调栈的模型

85. 最大矩形. 可转化为84题. 如下图所示思路： 时间复杂度$O(n^2)$.



柱子表示以该行为下边界，最多有多少个连续的1.

86. 分隔链表：可视作快排的一部分. 具体方法见下：

例：1→4→3→2→5→2. x=3.

新开两个链表：左→1→2→2; 右→4→3→5 ↪ 左尾指向右即可.

87. 扰乱字符串：注意是递归进行的算法步骤！ 方法一：爆搜(会超时)

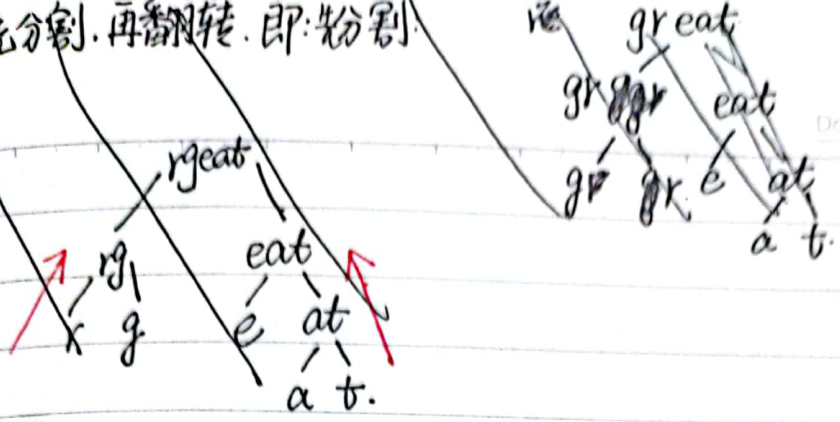
例如：great. 分割 → 递归：gr | eat. 交换 → rg | eat. 得：rgeat.

该部分递归过程略.

流程：
- 分割
- 交换or不交换.
- 递归~~子串~~分割后的子串进行上述两步骤.

实际上，可先分割，再翻转，即：分割

后翻转：



利用递归搜索求解，详见代码，注意代码中sort部分是一个剪枝，若不剪枝会超时。

时间复杂度：$a_n = 4(a_1 + a_2 + \cdots + a_{n-1})$ 又 $a_{n-1} = 4(a_1 + a_2 + \cdots + a_{n-2})$

$\therefore a_n - a_{n-1} = 4a_{n-1}$ $\therefore a_n = O(5^n)$

方法二：dp(区间dp)

i, j, k分别代表：S_1起始位置，S_2起始位置，考查的串的长度

①状态表示：$f(i,j,k)$ 表示将 $S_1[i, i+k-1]$ 转化成 $S_2[j, j+k-1]$ 的方案的集合是否非空。

②状态计算：若要计算 $f(i,j,k)$，只需考虑两种情况（遍历 u，$u \in [1, k-1]$）

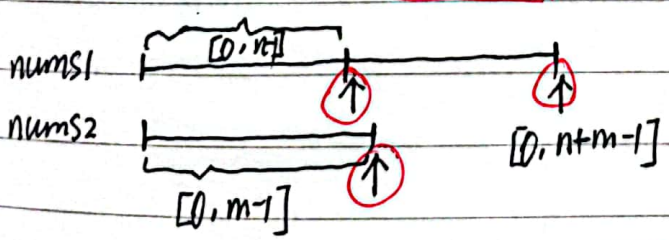
(i) i 起 u 长 $k-u$；j 起 u 长 $k-u$ 即只需判断 $f(i,j,u)$ 及 $f(i+u, j+u, k-u)$ → &&

(ii) i 起 u 长 $k-u$；j 起 $k-u$ 长 u 即只需判断 $f(i, j+k-u, u)$ 及 $f(i+u, j, k-u)$ → &&

两者 ||

空间复杂度：$O(n^3)$，时间复杂度：需考查 i, j, k, u（遍历）∴为 $O(n^4)$

88. 合并两个有序数组：注意由于nums1数组已足够容纳两数组中所有元素，因此不需开数的数组，对应的应从后往前遍历。



三指针算法，详见代码。

89. 格雷编码：其生成有一定规律，是递归生成的：

举例：n=0时 0 [1]序

n=1时，先复制一份n=0的答案：

补0 0 0

补1 1 1

n=2时，先倒序复制一份n=1的答案

补0 0 0 1

补1 1 1 0

n=0时，0

n=1时，0
1

n=2时，翻转复制：补0{00, 10；补1{11, 01

n=3时，翻转复制：补0{000, 100, 110, 010；补1{011, 111, 101, 001

90. 子集II：由于有重复元素，因此不能像78题二进制枚举，而要递归枚举。
思路：统计出每个数字有多少个，按~~数字的~~该数字在子集中出现的次数进行分类搜索。

91. 解码方法：dp问题 类似台阶问题。