动态规划：

L121-股票问题：



思路：  
（1）普通的低效率想法是两层循环进行比较，这样时间复杂度就是N平方了。  
（2）看到这题是动态规划，主要就是找到数组中最小的那个和其后面最大的那个数做差就是结果了，所以数组中做一次循环做两件事：第一件事，如果后面有更小的数据就跟新最小的数即代码中if(prices[i] < prices[j])  
 {j = i;}；第二件事：找最大的差，即代码if(compare >maxProf){  
 maxProf = compare;  
 } 。  
（3）对于边界问题，prices.size()==0或1，则循环不发生，直接返回0.

二、搜索类

阿里巴巴编程测试题：



思路：这题我总结下，思路有5种：1、就是我发的这种 ，大致就是我用了一个pair的vector进行存储，然后拿到你要搜索的关键字，用gbk编码格式编译，得到其字符的encode码，相同的汉字对应的码号是一样的，类似ASCII码中A-33，B-34，然后加到一起总和相等，去搜原来的存储空间，编程查找问题了，这里可以优化速度，我没有优化，我想测试其他的方式；2、用正则匹配去做，但是存在字个数不同需要控制的情况；3、走偏路，把查找的字符串全部组合罗列一个数组然后去匹配，这种也是没办法的办法。4、用Python的dict做，dict和顺序是无关的，只和出现的字和出现字数相关，比如你输入: 今天天气真的好好啊 那么dict[今] = 1, dict[天] = 2,.....。之后无论你输入 天气真的好好啊今天，都能算出来很相似。但是奈何人家不用Python。5、JAVA的今天没做完，不好说能不能整出来，理论上也是可以的，就是效率不知道咋样，弄出来给个结论。

三。最大子串类问题【引入后缀数组】

B8.23华为-后缀数组求最长子串

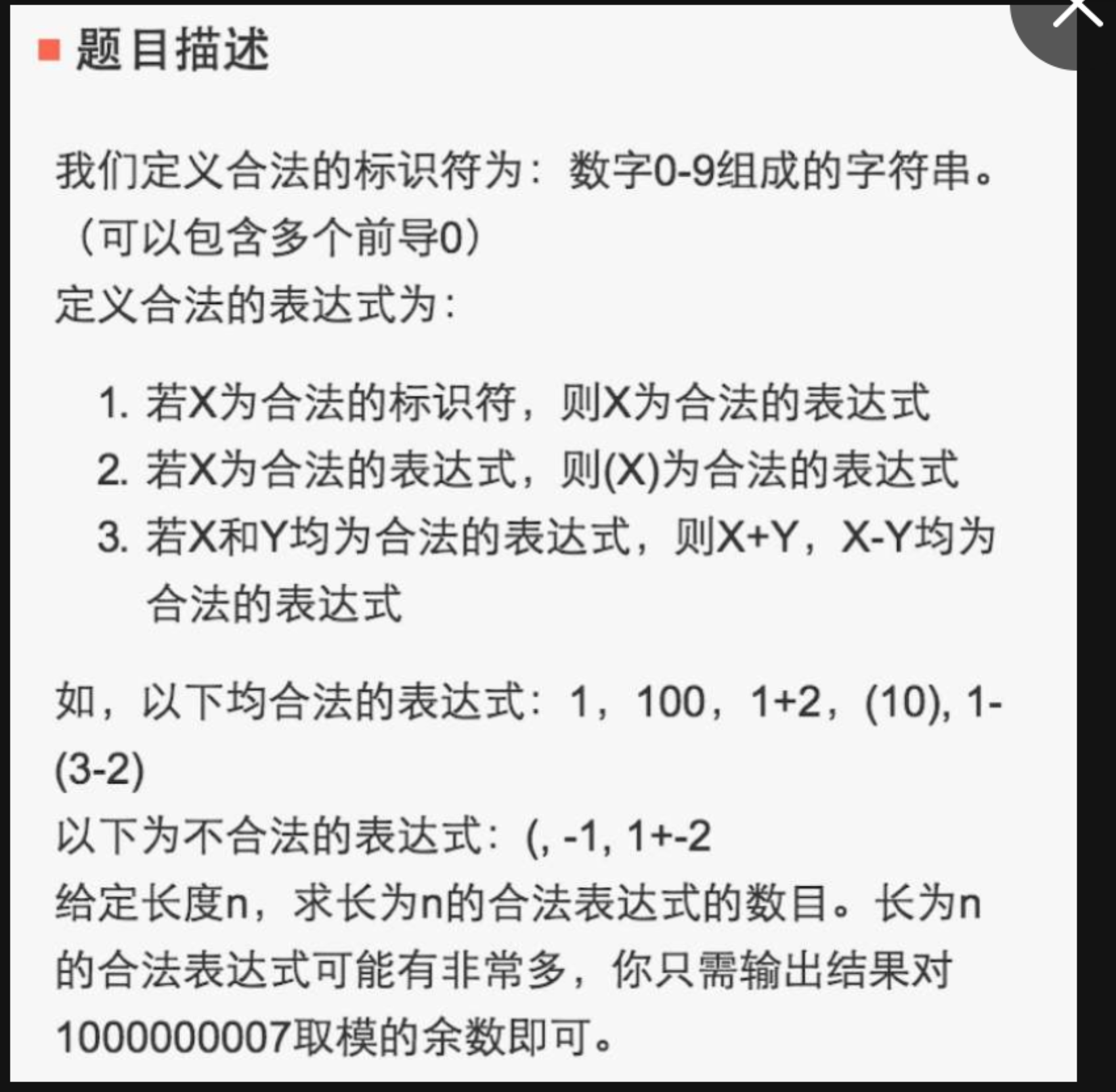
<https://blog.csdn.net/Bule_Zst/article/details/78604864>

<https://www.baidu.com/link?url=bd_5zcTn0dw4cstxw6YTn6w6IH5JkFqk_pEh7GSuSV58GDCj-xkbehGxCqhC07LHxyNEC_E0I_shtk-7bDWeLK&wd=&eqid=c5304057000c82f9000000035b82cb32>

基本就是第一篇博客看懂就明白后缀数组了。

四、动态规划

B8.25头条-合法表达式



解题思路：

