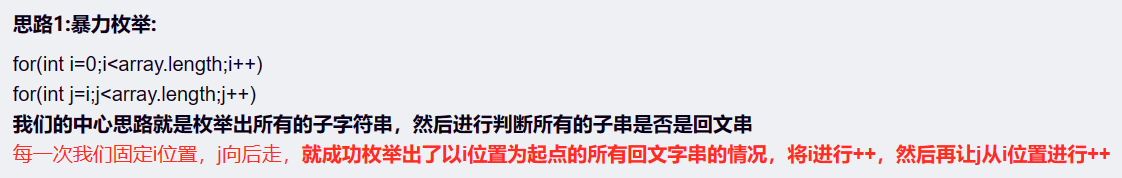
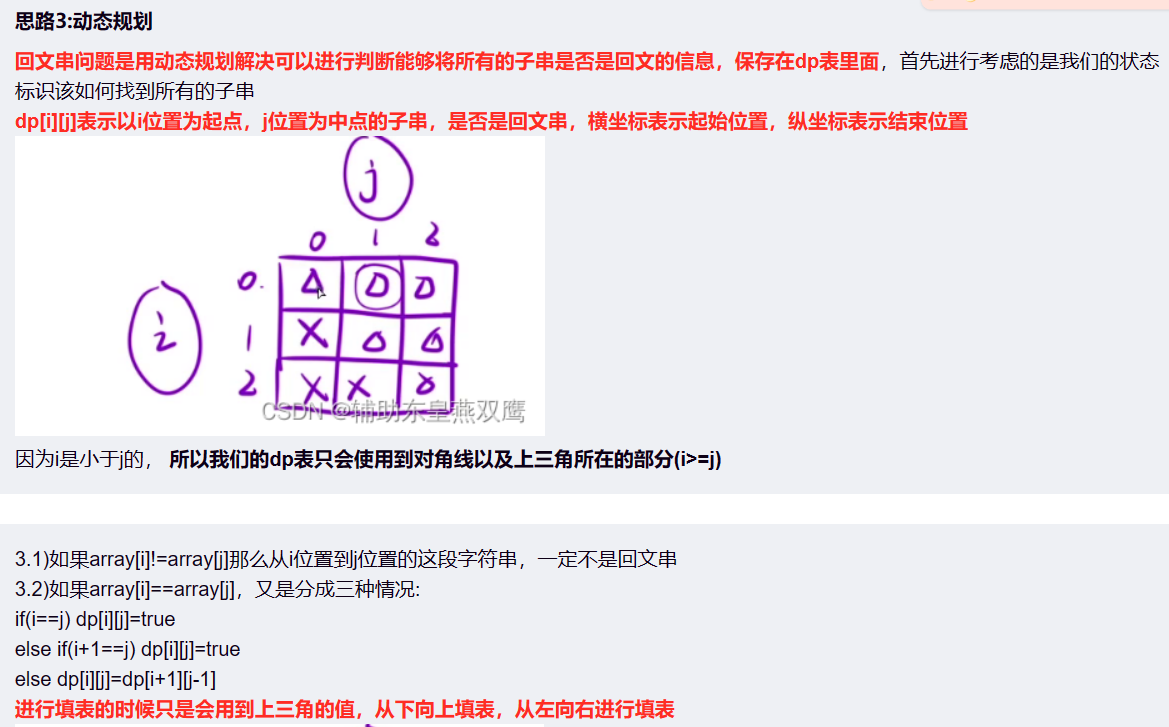
[动态规划--回文串问题\_回文字符串动态规划\_写个堆排的博客-CSDN博客](https://blog.csdn.net/weixin_61518137/article/details/131872359)

<https://blog.csdn.net/weixin_61518137/article/details/131872359>







# 思路三：动态规划核心点讲解

输入子串s = **bab**

状态转移方程：f(i,j)表示s中从i到j的子串是否为回文串

1. s[i]!=s[j]，f(i,j)=false（因为从i到j的这段子串中，s[i]和s[j]这两个对称边界值都不一样，肯定不是回文串）

2. s[i]==s[j]，

(1) i==j，f(i,j)=true（因为当前只有一个字母，肯定为回文串）

(2) i+1 == j, f(i,j)=true（因为当前是相邻的两个字母，相同的话肯定为回文串）

(3) otherwise, f(i+1, j-1)（**i+1表示起点字符向后移动1位，j-1表示终点字符向前移动1位，f(i+1, j-1)整体表示夹在i和j中间的子串是否是回文串**）

填写dp表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | b | a | b |
|  | **j**  **i** | 0 | 1 | 2 |
| b | 0 | √b | ×ba | √bab |
| a | 1 | 空 | √a | ×ab |
| b | 2 | 空 | 空 | √b |

注意：

1. 初始化: 不用初始化
2. 填表顺序: **从下向上，从左向右，**每一行，因为当进行填写当前行的时候要使用到下一行的值。
3. 只用到上三角区域的值，因为i<=j