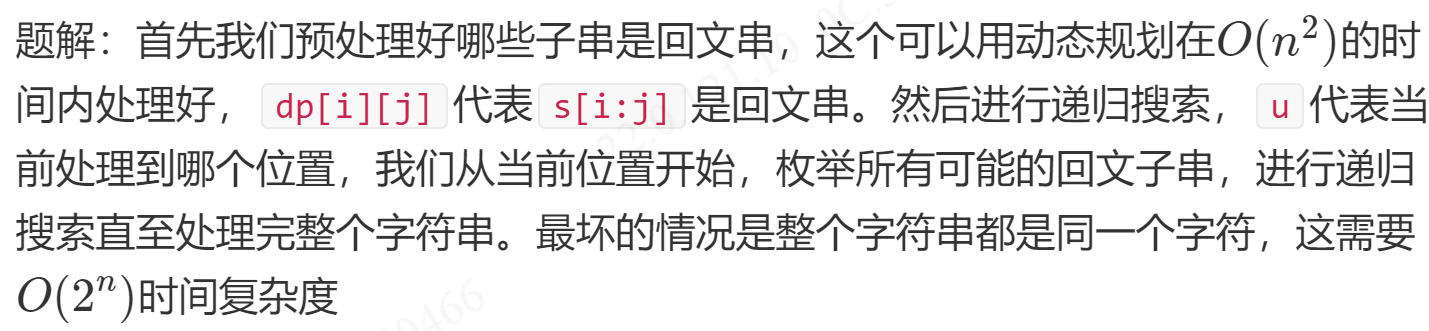
**131. Palindrome Partitioning**

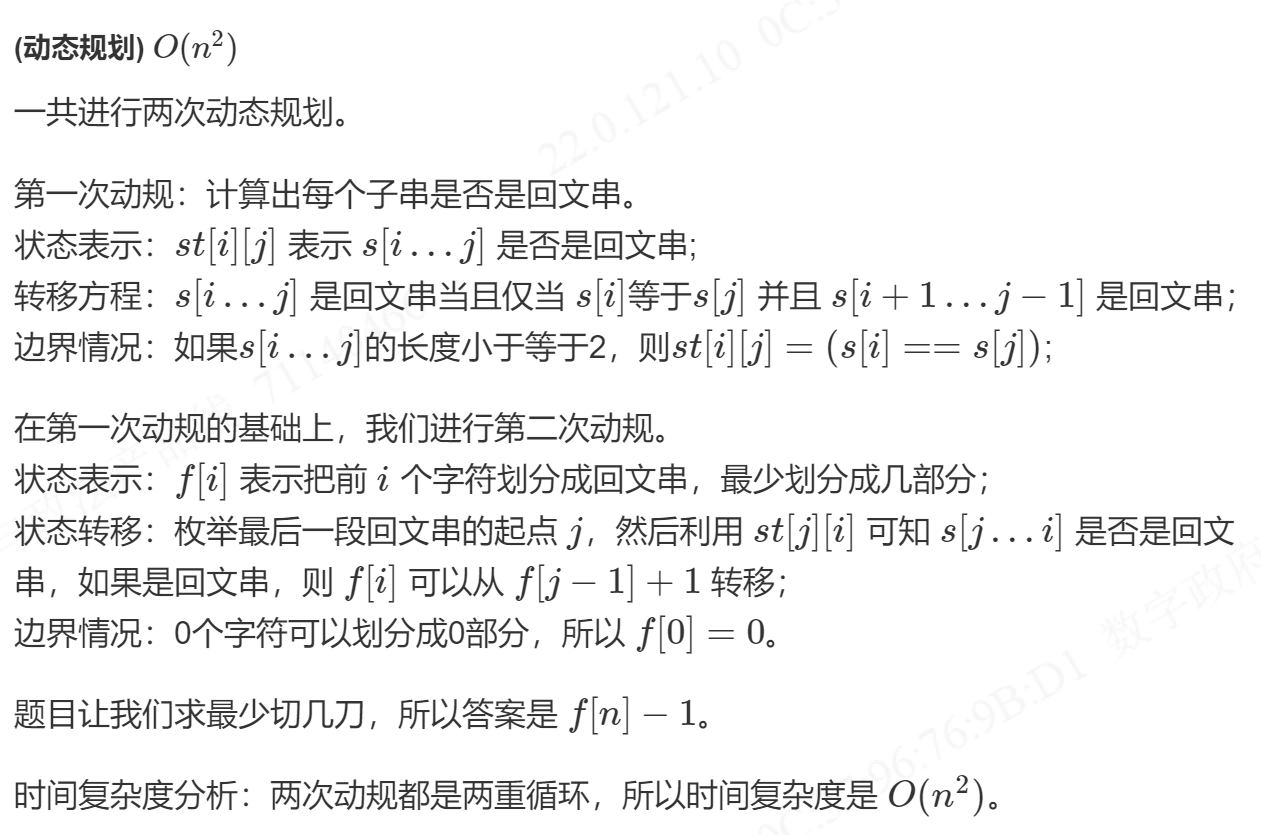


注意下C++的一个语法：

s.substr(pos, len): pos表示要截取的子字符串在字符串s中的起始位置，len表示要截取的子字符串的长度。

**132.Palindrome Partitioning II**

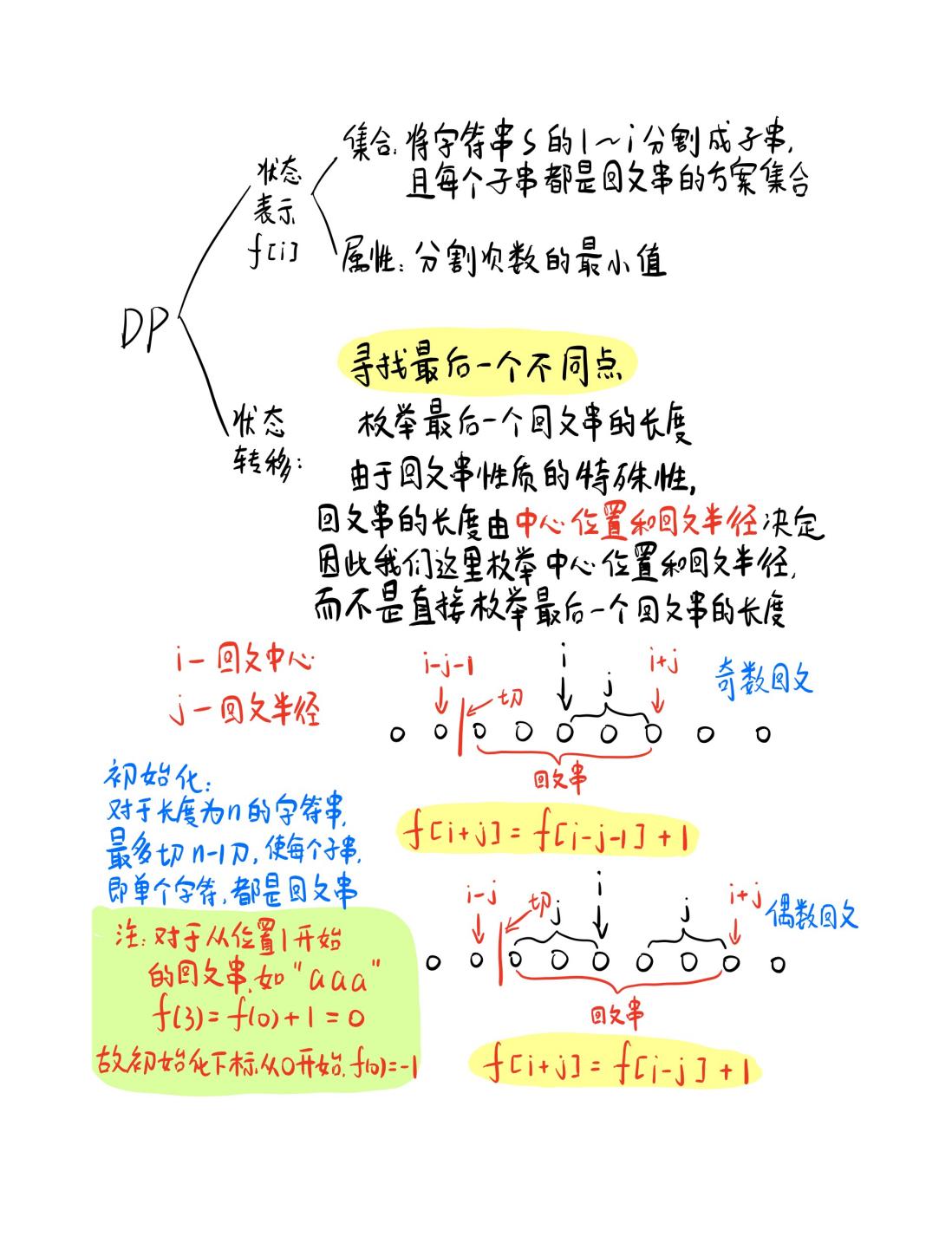
**解法一：**



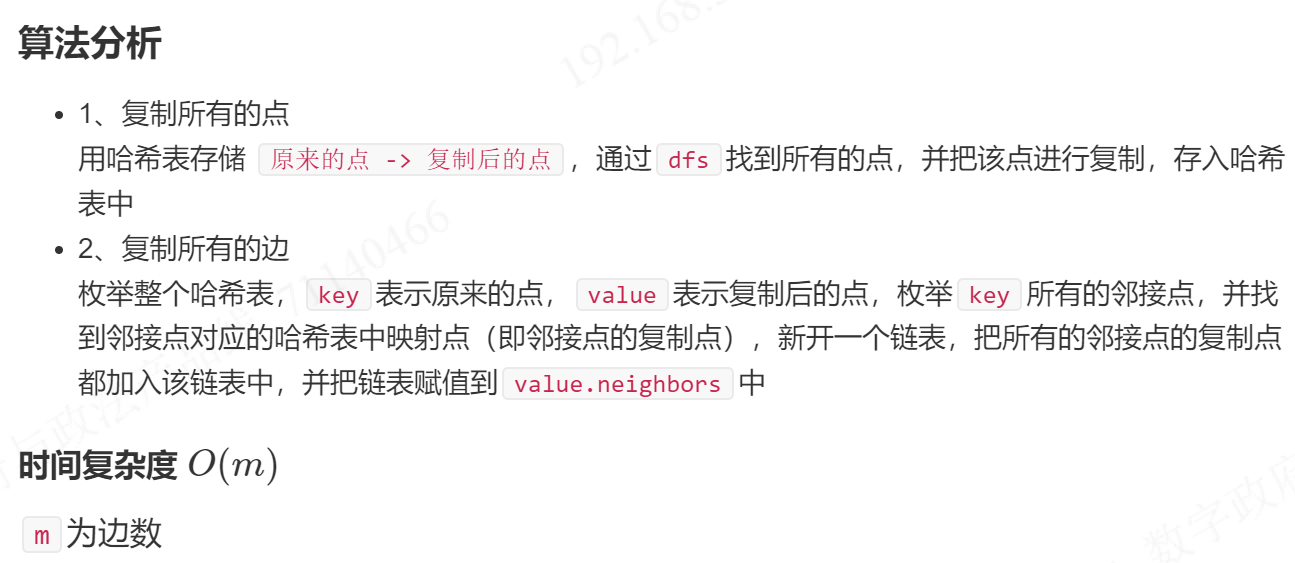
**解法一**如上。进行两次动态规划即可。此方法在第一次动归时开的空间比较多。

**解法二：**

与解法一相比，利用的空间更少。



**133.Clone Graph**



C++实现语法细节：

1.unordered\_map

①count函数：由于unordered\_map不允许存储具有重复键的元素，因此count()函数本质上检查unordered\_map中是否存在具有给定键的元素。如果Map中存在具有给定键的值，则此函数返回1，否则返回0。

②插入元素：可以用下标索引的形式插入元素，语法为unordered\_map[key] = value，如果键 key不存在，则插入键和对应的值，否则，会将键key的值更新为value。

③遍历：c++17中可以用**结构化绑定**的方式遍历哈希表，语法形式：auto [k,v]:map。

2.vector

void push\_back(const T& x):向量尾部增加一个元素x

**134.Gas Station**

使用**贪心(双指针)**算法，时间复杂度为O(n)，空间复杂度为O(1).

利用如下结论：

如果从x最远能到y点，y + 1点到不了，那么从x到y的任一点k出发都不可能到达y+1。

证明:

因为从k点出发的话，相当于从0开始加油，而如果从x出发到k点的油量一定是>=0的，即还有剩余的油或正好没油。既然在k点时油量>=0都到不了y+1，那么从0开始就更不可能到达y+1了。

**135.Candy**

方法一：两次遍历，时间复杂度O(n)，空间复杂度O(n)。