

Leetcode 6003.

相似题目:

1. 给一个数组, 只能从两边取数, 一共取K个, 要和最大
2. 给一个包含"A","B"的字符串, 可以从两端移除字符, 代价是1, 也可以从中间直接移除字符, 代价是2. 如何花费最少代价得到全部是A的字符.

TLE解法:

1. 回溯

backtracking(S, time)

2. 二维dp.

$$dp[i][j] = \min(dp[i+1][j] + 1, dp[i][j-1] + 1, 2 \times \text{num of } 1)$$

动规五步曲:

①. 确定dp数组大小, 以及下标含义:

$\begin{cases} dp_prefix: \text{size } n, i \text{ 表示只用操作1得到合法字符串} \\ dp_suffix: \text{size } n, i \text{ 表示只用操作2, 3得到合法字符串.} \end{cases}$

②. 递推公式:

$\begin{cases} dp_prefix[i] = dp_prefix[i-1] + diff * s[i] \\ dp_suffix[i] = \min(dp_suffix[i+1] + 2, n-i) \end{cases}$

③. 初始化:

$\begin{cases} dp_prefix[0] = s[0] \\ dp_suffix[n-1] = s[n-1] \end{cases}$

④. 遍历顺序.

$dp_prefix: 0, 1, 2, 3, \dots, n$

$dp_suffix: 0, 1, 2, 3, \dots, n$

⑤. 返回合适的值.

$\text{return } \min(dp_prefix[i] + dp_suffix[i+1])$