动态规划习题课

一、海贼 OJ-46-切割回文

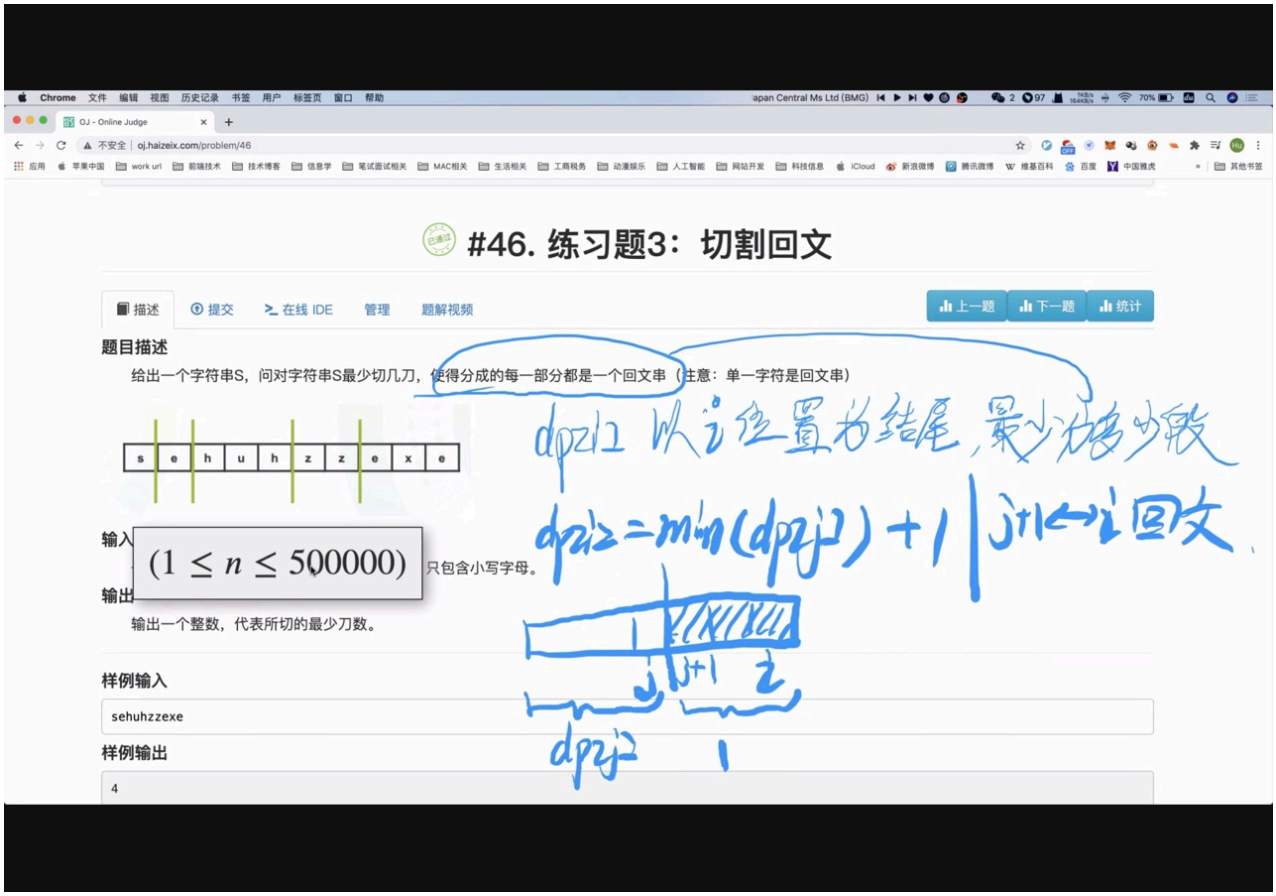
状态定义

$dp[i]$ 代表取字符串的前 i 位，最少分成多少段回文串

状态转移

$dp[i] = min(dp[j]) + 1 \mid s[j+1, i]\ is\ palindrome$

1. 根据状态转移，算法时间复杂度$O(n^2)$
2. 所以，我们需要对转移阶段进行优化
3. 动态规划优化章节的时候，重点解决



二、海贼-OJ-47-01背包

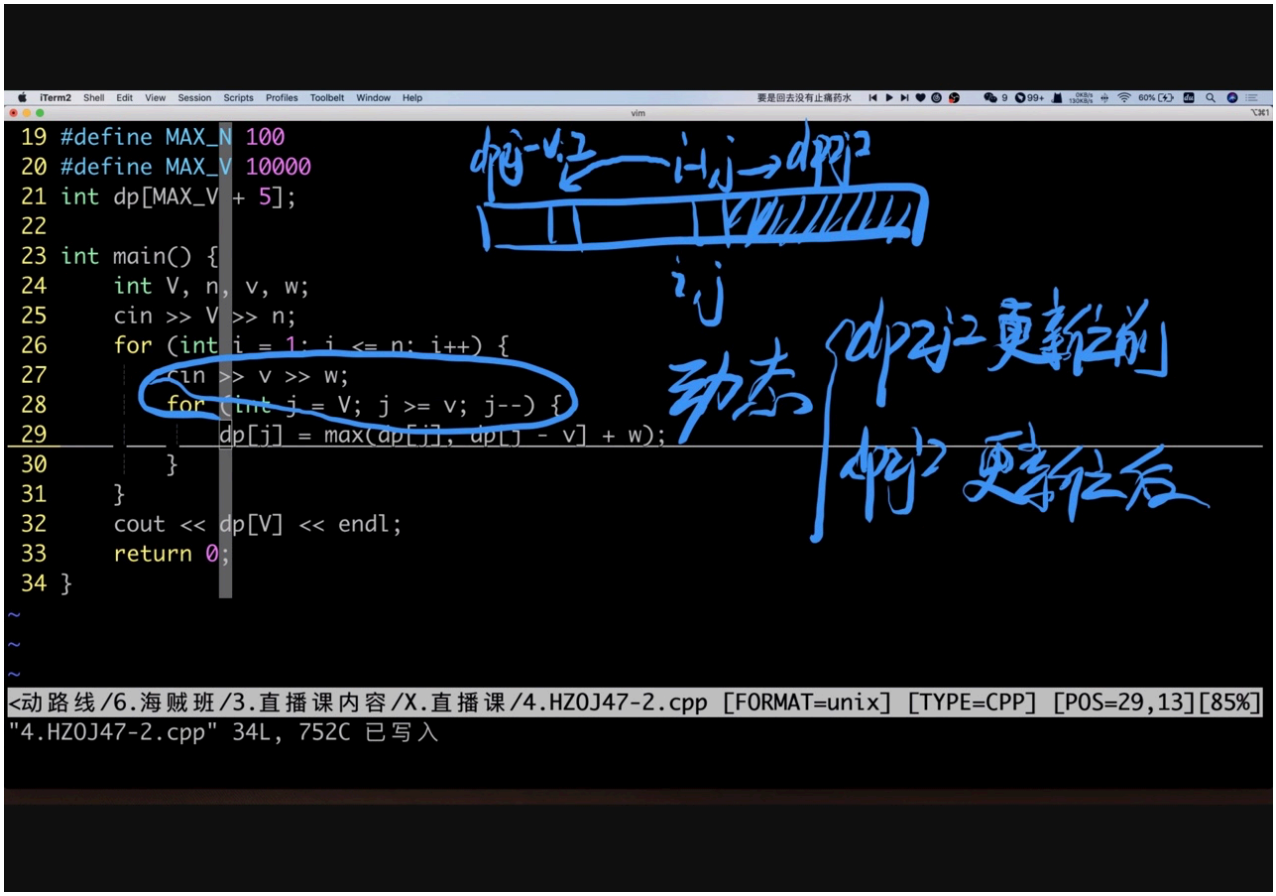
状态定义

$dp[i][j]$ 前 i 个物品，背包最大承重为 j 的情况下，最大价值

状态转移

$$dp[i][j] = max \begin{cases} dp[i-1][j] & \text{没选第}i\text{件} \\ dp[i-1][j-v[i]] + w[i] & \text{选了第}i\text{件} \end{cases}$$

1. 第一种程序实现，状态如何定义的，程序就如何实现
2. 第二种程序实现，使用滚动数组，对代码进行了空间优化
3. 第三种程序实现，将程序中的 dp 数组变成一维的，并且修改了更新顺序



三、海贼 OJ-48-完全背包

状态定义

$dp[i][j]$ 前 i 个物品，背包最大承重为 j 的情况下，最大价值

状态转移

$$dp[i][j] = max \begin{cases} dp[i-1][j] & \text{没选第}i\text{件} \\ dp[i][j-v[i]] + w[i] & \text{选了若干个第}i\text{件} \end{cases}$$

程序实现的时候，参考01背包的程序实现，将逆向刷表，改成正向刷表

四、海贼 OJ-49-多重背包

问题模型转换

1. 多重背包，每类物品多了一个数量限制
2. 01背包，每种物品只有一个
3. 将多重背包中的数量限制，当做多个单一物品来处理
4. 至此就将多重背包，转成了0/1背包问题

状态定义

$dp[i][j]$ 前 i 个物品，背包最大承重为 j 的情况下，最大价值

状态转移

$$dp[i][j] = max \begin{cases} dp[i-1][j] & \text{没选第}i\text{件} \\ dp[i-1][j-v[i]]+w[i] & \text{选了第}i\text{件} \end{cases}$$

五、海贼 OJ-50-扔鸡蛋

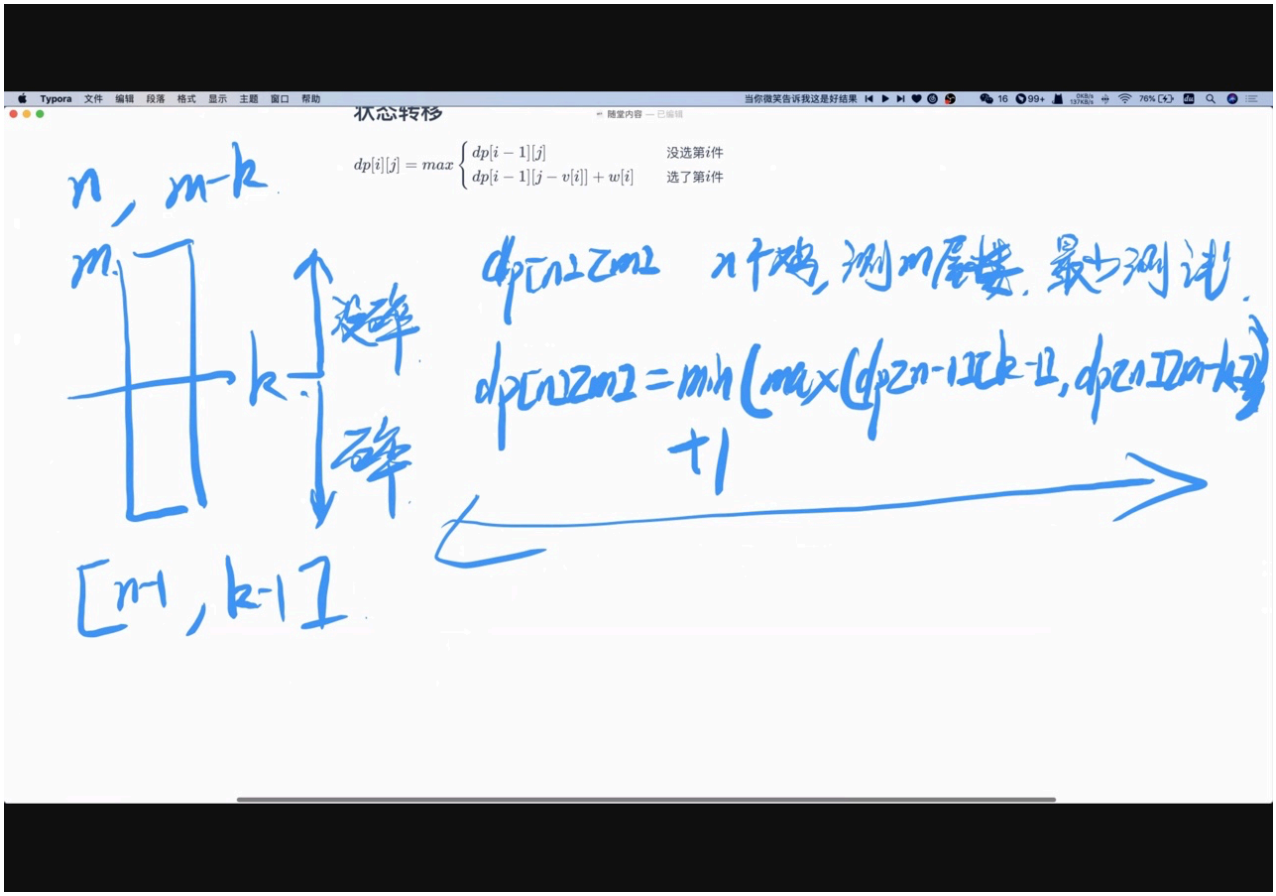
状态定义

$dp[n][m]$ 用 n 个鸡蛋，测 m 层楼，最坏情况下最少测多少次

状态转移

$$dp[n][m] = min(max \begin{cases} dp[n-1][k-1]+1 & \text{鸡蛋碎了} \\ dp[n][m-k]+1 & \text{鸡蛋没碎} \end{cases})$$

1. 程序所使用的存储空间与楼层数量强相关
2. 楼层数量达到了 2^{31}，所以在这种状态定一下不可行
3. 状态定义不可行，我们就需要优化状态定义
4. 时间复杂度 $O(n \times m^2)$，当 m 过大的时候，无法通过时间限制


状态转移
$dp[i][j] = max \begin{cases} dp[i-1][j] & \text{没选第}i\text{件} \\ dp[i-1][j-v[i]]+w[i] & \text{选了第}i\text{件} \end{cases}$
n, m-k
m
k
选择
放弃
[n-1, k-1]
dp[n][m] n个物, 到m层楼. 最小测试.
dp[n][m] = min(max(dp[n-1][k-1], dp[n][m-k]) +1