重庆南开中学

动态规划 III

南开中学信息学竞赛教练组

目录
contents

PART 01

作业评讲

- 讲

作业3636 – 三个序列的

题目大意：

输入三个整数序列，求它们的最长公共子序列长度。每个序列长度不超过200。

解析：

- 设定状态：f[i][j][k]表示x[1..i],y[1..j],z[1..k]的最长公共子序列长度
- 状态转移方程：

$$f[i][j][k] = \begin{cases} f[i-1][j-1][k-1]+1 & \texttt{if}\ \ x[i]=y[j]=z[k] \\ \max(f[i-1][j][k], f[i][j-1][k], f[i][j][k-1]) & \texttt{other} \end{cases}$$

- 边界情况：i,j,k至少一个等于0时，f[i][j][k]=0
- 最终答案：f[X][Y][Z]
- 优化：f数组可以将最外层循环的维度省略。

作业2658 – 最长公共子序列

题目大意：

输入两个只含大写字母的字符串，求最长公共子序列的长度和数量。字符串长度⩽5000。

解析：

- 设定状态：
  - f[i][j]表示a[1..i]和b[1..j]的最长公共子序列长度
  - g[i][j]表示a[1..i]和b[1..j]的最长公共子序列数量
- 状态转移：
  - 先把f[i][j]算出来，并令g[i][j]=0
  - 如果f[i][j]==f[i-1][j]，则g[i][j]+=g[i-1][j]
  - 如果f[i][j]==f[i][j-1]，则g[i][j]+=g[i][j-1]
  - 如果f[i][j]==f[i-1][j-1]，刚才加的两部分有重复，g[i][j]需要减去重复的g[i-1][j-1]
  - 如果a[i]==b[j]，则g[i][j]还要加上g[i-1][j-1]
- 时间复杂度$O(n^2)$

PART 02

矩阵上的DP

- 都比较简单

例题

例题1：【NKOJ1182 建别墅】

何老板买了一块面积为$n \times m$的土地，他想在这块土地上建造一座别墅。按照中国传统四平八稳的思想，他希望这个别墅是正方形的。但是，这块土地并非十全十美，上面有很多地方是不平坦的，以至于根本不能在上面盖一砖一瓦。他希望找到一块最大的平坦的正方形土地来盖别墅。应该选哪一块土地呢？现在，你来告诉他吧。

输入：

第一行为两个整数$n, m(1 \leq n, m \leq 1000)$。
接下来n行，每行m个数字，用空格隔开。0表示该块土地不平坦，1表示该块土地平坦。

输出：

一个整数，最大正方形的边长。

样例输入：

```
4 4
0 1 1 1
1 1 1 0
0 1 1 0
1 1 0 1
```

样例输出：

```
2
```

例题

例题1：【NKOJ1182 建别墅】

解析：

- 设定状态：f[i][j]表示以点(i,j)作为正方形右下角的正方形最大边长。

例题

例题1：【NKOJ1182 建别墅】

解析：

- 设定状态：f[i][j]表示以点(i,j)作为正方形右下角的正方形最大边长。
- 转移：
  - 如果a[i][j]＝0，则f[i][j]=0。
  - 如果a[i][j]＝1，f[i][j]怎么计算？和其他状态有什么关系？

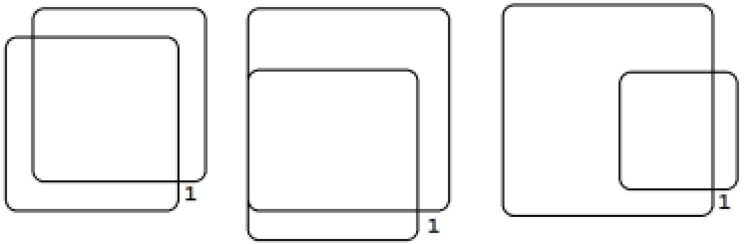
重庆南开中学

例题

例题1：【NKOJ1182 建别墅】

解析：

- 设定状态：f[i][j]表示以点(i,j)作为正方形右下角的正方形最大边长。
- 转移：
  - 如果a[i][j]＝0，则f[i][j]=0。
  - 如果a[i][j]＝1，f[i][j]怎么计算？和其他状态有什么关系？
  - f[i][j]可以等于min(f[i-1][j]，f[i][j-1])
  - f[i][j]能否等于min(f[i-1][j]，f[i][j-1]) + 1呢？
    - 需要f[i-1][j-1]⩾min(f[i-1][j]，f[i][j-1])



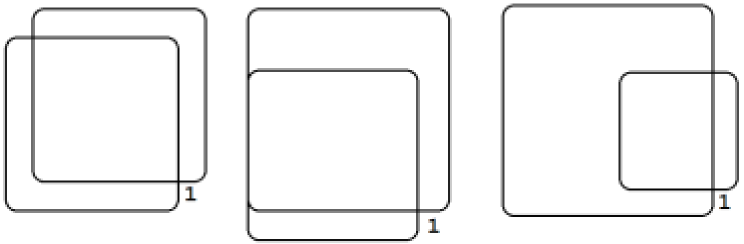
例题

重庆南开中学

例题1：【NKOJ1182 建别墅】

解析：

- 设定状态：f[i][j]表示以点(i,j)作为正方形右下角的正方形最大边长。
- 转移：
  - 如果a[i][j]＝0，则f[i][j]=0。
  - 如果a[i][j]＝1，f[i][j]怎么计算？和其他状态有什么关系？
  - f[i][j]可以等于min(f[i-1][j]，f[i][j-1])
  - f[i][j]能否等于min(f[i-1][j]，f[i][j-1]) + 1呢？
    - 需要f[i-1][j-1]≥min(f[i-1][j]，f[i][j-1])
    - 注意到f[i-1][j-1]≥min(f[i-1][j]，f[i][j-1]) - 1
  - 所以，f[i][j] = min(f[i-1][j]，f[i][j-1]，f[i-1][j-1]) + 1。



例题

例题1：【NKOJ1182 建别墅】

解析：

- 设定状态：f[i][j]表示以点(i,j)作为正方形右下角的正方形最大边长。
- 状态转移方程：

$$f[i][j]=\begin{cases}0 & \texttt{if}\ \ a[i][j]=0\\ \min(f[i-1][j],f[i][j-1],f[i-1][j-1])+1 & \texttt{if}\ \ a[i][j]=1\end{cases}$$

- 边界情况：i=0或j=0时，f[i][j] = 0
- 最终答案：f数组中最大的数
- 时间复杂度$O(n^2)$。

例题

例题1：【NKOJ1182 建别墅】

思维发散：

- 如果建别墅的区域是矩形，该怎么做？

例题

例题1：【NKOJ1182 建别墅】

思维发散：

- 如果建别墅的区域是矩形，该怎么做？
- 讲一个非DP的算法：
- 如果矩形一定在整个区域的底部，就和单调队列例题“广告牌”完全相同了！
  - 当别墅左右边界固定时，上下要尽量高，则上方一定会顶住某个0；
  - 枚举顶住哪一列的0，计算左右最远可以扩展多远。

例题

例题1：【NKOJ1182 建别墅】

思维发散：

- 如果建别墅的区域是矩形，该怎么做？
- 讲一个非DP的算法：
- 如果矩形一定在整个区域的底部，就和单调队列例题“广告牌”完全相同了！
  - 当别墅左右边界固定时，上下要尽量高，则上方一定会顶住某个0；
  - 枚举顶住哪一列的0，计算左右最远可以扩展多远。
- 假设别墅矩形的下边界是第i行
  - 设第j列从下边界起有b[j]个连续的1，即a[i][j]到a[i-b[j]+1][j]都是1
  - b[j]可以用从上往下的前缀和计算出来

例题

例题1：【NKOJ1182 建别墅】

思维发散：

- 如果建别墅的区域是矩形，该怎么做？
- 讲一个非DP的算法：
- 如果矩形一定在整个区域的底部，就和单调队列例题“广告牌”完全相同了！
  - 当别墅左右边界固定时，上下要尽量高，则上方一定会顶住某个0；
  - 枚举顶住哪一列的0，计算左右最远可以扩展多远。
- 假设别墅矩形的下边界是第i行
  - 设第j列从下边界起有b[j]个连续的1，即a[i][j]到a[i-b[j]+1][j]都是1
  - b[j]可以用从上往下的前缀和计算出来
  - 用单调队列找出b[j]左边和右边第一个比自己小的数b[j1]和b[j2]
  - 则下边界为第i行、上面第j列被顶住时，矩形最大宽度为j2-j1-1，更新答案
- 时间复杂度$O(n^2)$

例题

例题2：【NKOJ2553 开垦农田】

何老板买了一块土地，他想在这块土地上开垦出一块矩形的农田。这块土地可看做由n*n个小方块土地构成，何老板告诉你每小块土地的肥沃值，他希望开垦出一片肥沃值总和最大的矩形农田。
问：这个最大肥沃值总和是多少？
数据范围：$1 \le n \le 100$，每个数绝对值不超过10000

输入：

第一行为整数n。接下来是一个由整数构成的n*n的矩阵。

输出：

一个整数，表示所求答案。

样例输入：
4
0 -2 -7 0
9 2 -6 2
-4 1 -4 1
-1 8 0 -2

样例输出：
15

例题

例题2：【NKOJ2553 开垦农田】

解析：

- 简化问题：
  - 如果开垦区域的上下边界固定，如何计算最优的左右边界？

例题

例题2：【NKOJ2553 开垦农田】

解析：

- 简化问题：
  - 如果开垦区域的上下边界固定，如何计算最优的左右边界？
  - 算出上下边界内每列所有数的和，设第i列的和为s[i]
  - 问题转化为求s[]数组的最大连续和，时间复杂度$O(n)$

重庆南开中学

例题

例题2：【NKOJ2553 开垦农田】

解析：

- 简化问题：
  - 如果开垦区域的上下边界固定，如何计算最优的左右边界？
  - 算出上下边界内每列所有数的和，设第i列的和为s[i]
  - 问题转化为求s[]数组的最大连续和，时间复杂度$O(n)$
- 枚举上下边界，如何计算每次的s[]数组？
  - 对原数组a[][]从上往下求前缀和u[][]，
  - 上下边界分别为第x行和第y行时，第i列的和s[i]=u[y][i]-u[x-1][i]
- 总时间复杂度$O(n^3)$

PART 03

区间类DP

- 也不难

例题

例题3：【石子合并】

N堆石子摆放成一个圆形，现要将石子有次序地合并成一堆。每次只能选相邻的2堆合并成新的一堆，并将新的一堆的石子数记为该次合并的得分。设计出一个算法，计算出将N堆石子合并成1堆的最小得分和最大得分。

数据范围：$N \le 100$

输入：

第1行是正整数N，表示有N堆石子。第2行有N个数，分别表示每堆石子的个数。

输出：

输出共2行，第1行为最小得分，第2行为最大得分。

样例输入：	样例输出：
4	43
4 4 5 9	54

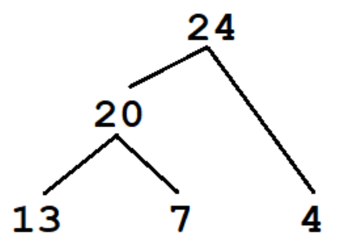
例题

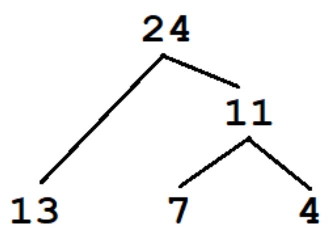
例题3：【石子合并 简化版】

N堆石子摆放成一列，现要将石子有次序地合并成一堆。每次只能选相邻的2堆合并成新的一堆，并将新的一堆的石子数记为该次合并的得分。计算出将N堆石子合并成1堆的最小得分。

数据范围：$N \leq 100$

例如N=3堆，分别有13,7,4个石子。合并方法有两种，如右图所示。两种方法得分分别是44和35，所以最小得分是35。



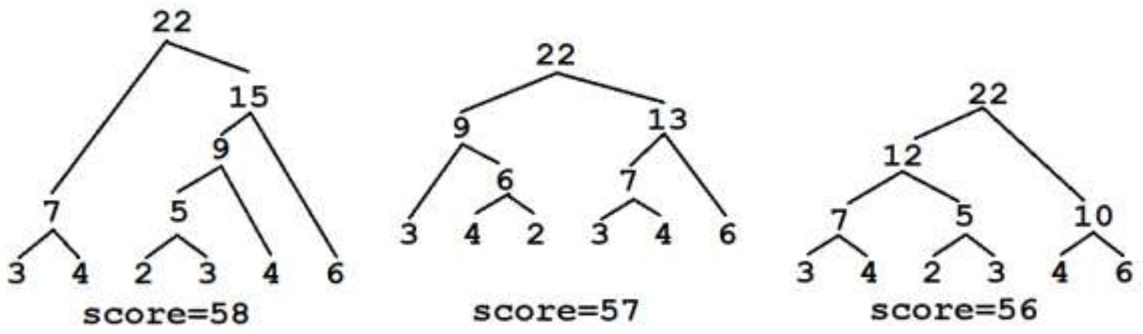


例题

例题3：【石子合并 简化版】

解析：

- 脑补一个贪心策略：
  - 每次合并总数最小的相邻两堆
- 脑补另一个贪心策略：
  - 每次在中间尽量均匀划分成两堆，递归
  - 分治两边得到两堆，然后合并这两对
- 然后举出一个反例



例题

例题3：【石子合并 简化版】

解析：

- 动态规划：如何设定状态？

例题

例题3：【石子合并 简化版】

解析：

- 动态规划：如何设定状态？
- 多阶段问题化为单阶段问题
  - 多阶段问题：将n堆石子合并成一堆
  - 每个阶段：将若干堆石子合并成一堆
  - 单阶段问题：将若干堆石子合并成两堆之后，最后再合并成一堆

重庆南开中学

例题

例题3：【石子合并 简化版】

解析：

- 动态规划：如何设定状态？
- 多阶段问题化为单阶段问题
  - 多阶段问题：将n堆石子合并成一堆
  - 每个阶段：将若干堆石子合并成一堆
  - 单阶段问题：将若干堆石子合并成两堆之后，最后再合并成一堆
- 设定状态：
  - f[i][j]表示：把从第i堆石子起的连续j堆石子合并成一堆的最小得分
  - 其中$1 \le i \le n$，$1 \le j \le n - i + 1$

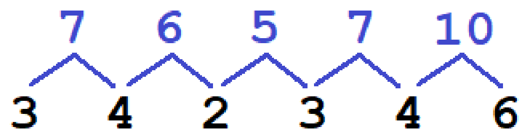
例题

例题3：【石子合并 简化版】

解析：

- 动态规划：如何转移？
- 边界情况：j=1时，本来就只有1堆，不需要合并，所以f[i][1]=0

例题



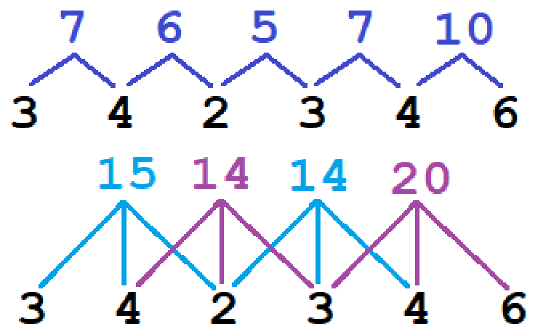
例题3：【石子合并 简化版】

解析：

- 动态规划：如何转移？
- 边界情况：j=1时，本来就只有1堆，不需要合并，所以f[i][1]=0
- 当j=2时：
  - 只有一种合并方式

$$f[i][2]=a_i+a_{i+1}+f[i][1]+f[i+1][1]$$

例题



例题3：【石子合并 简化版】

解析：

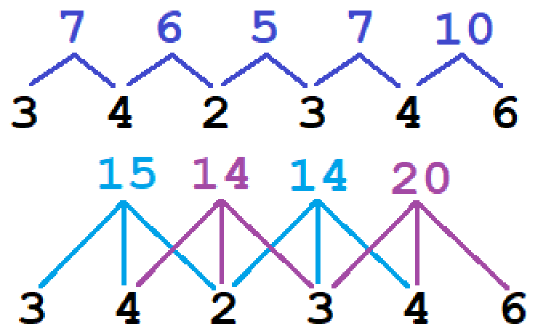
- 动态规划：如何转移？
- 边界情况：j=1时，本来就只有1堆，不需要合并，所以f[i][1]=0
- 当j=2时：
  - 只有一种合并方式

$$f[i][2] = a_i + a_{i+1} + f[i][1] + f[i+1][1]$$

- 当j=3时：
  - 可以先合并前2堆，也可以先合并后2堆

$$f[i][3] = a_i + a_{i+1} + a_{i+2} + \min\begin{cases} f[i][2] + f[i+2][1] \\ f[i][1] + f[i+1][2] \end{cases}$$

例题

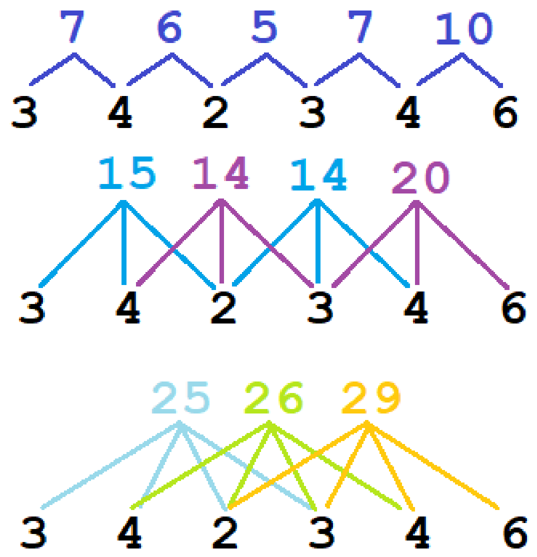


例题3：【石子合并 简化版】

解析：

- 动态规划：如何转移？
- 每次合并时，分值都要加上这段石子的总数
  - 不妨设sum(i,j)表示从第i堆石子起的连续j堆的总数

例题



例题3：【石子合并 简化版】

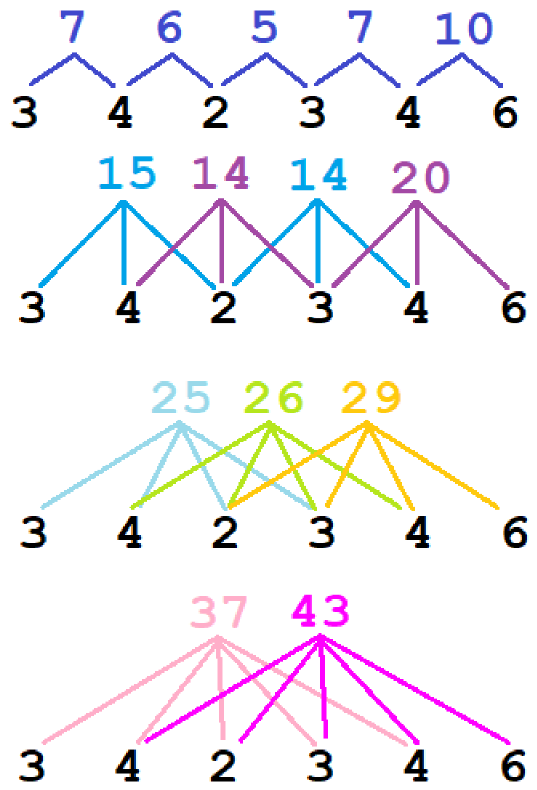
解析：

- 动态规划：如何转移？
- 每次合并时，分值都要加上这段石子的总数
  - 不妨设sum(i, j)表示从第i堆石子起的连续j堆的总数
- 当j=4时：
  - 可以按3+1、2+2、1+3三种方式合并：

$$f[i][4] = sum(i,4) + \min\begin{cases} f[i][3] + f[i+3][1] \\ f[i][2] + f[i+2][2] \\ f[i][1] + f[i+1][3] \end{cases}$$

例题

例题3：【石子合并 简化版】



解析：

- 动态规划：如何转移？
- 每次合并时，分值都要加上这段石子的总数
  - 不妨设sum(i, j)表示从第i堆石子起的连续j堆的总数
- 当j=4时：
  - 可以按3+1、2+2、1+3三种方式合并：

$$f[i][4] = sum(i,4) + \min \begin{cases} f[i][3] + f[i+3][1] \\ f[i][2] + f[i+2][2] \\ f[i][1] + f[i+1][3] \end{cases}$$

- 当j=5时：
  - 可以按4+1、3+2、2+3、1+4四种方式合并：

$$f[i][5] = sum(i,5) + \min \begin{cases} f[i][4] + f[i+4][1] \\ f[i][3] + f[i+3][2] \\ f[i][2] + f[i+2][3] \\ f[i][1] + f[i+1][4] \end{cases}$$

例题

例题3：【石子合并 简化版】

解析：

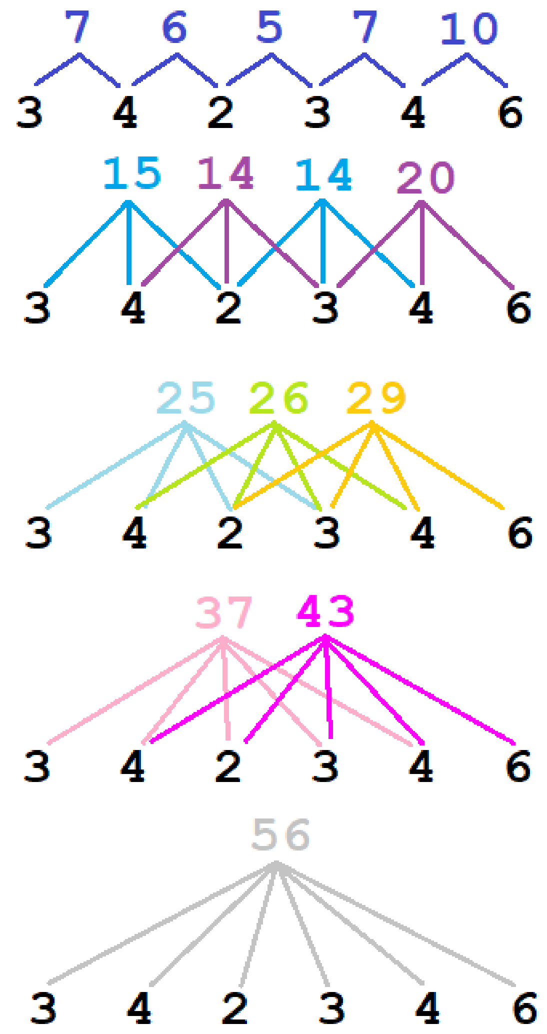
- 动态规划：如何转移？
- 每次合并时，分值都要加上这段石子的总数
  - 不妨设sum(i, j)表示从第i堆石子起的连续j堆的总数
- 当j=4时：
  - 可以按3+1、2+2、1+3三种方式合并：

$$f[i][4] = sum(i,4) + \min\begin{cases} f[i][3] + f[i+3][1] \\ f[i][2] + f[i+2][2] \\ f[i][1] + f[i+1][3] \end{cases}$$

- 当j=5时：
  - 可以按4+1、3+2、2+3、1+4四种方式合并：

$$f[i][5] = sum(i,5) + \min\begin{cases} f[i][4] + f[i+4][1] \\ f[i][3] + f[i+3][2] \\ f[i][2] + f[i+2][3] \\ f[i][1] + f[i+1][4] \end{cases}$$

- 当j=6时：
  - 略



例题

例题3：【石子合并 简化版】

解析：

- 动态规划：如何转移？
- 状态转移方程：
  - 将j≥2的所有情况统一起来
  - f[i][j]可以先将前k堆和后j-k堆分别合并成一堆，再将这两堆合并成一堆

$$f[i][j] = sum(i,k) + \min(f[i][k] + f[i+k][j-k]) \quad \texttt{for } 1 \le k \le j-1$$

例题

例题3：【石子合并 简化版】

解析：

- 动态规划：如何转移？
- 状态转移方程：
    - 将j⩾2的所有情况统一起来
    - f[i][j]可以先将前k堆和后j-k堆分别合并成一堆，再将这两堆合并成一堆

$$f[i][j] = sum(i,k) + \min(f[i][k] + f[i+k][j-k]) \quad \texttt{for}\ 1 \le k \le j-1$$

- 边界情况：f[i][1] = 0
- 最终答案：f[1][n]
- 时间复杂度$O(n^3)$

例题

例题3：【石子合并 ~~简化版~~】

解析：

- 回过神来！原题还没有解决！
- 最大得分？
    - 和最小得分相似，状态转移时min全部换成max即可
- n堆石子排成环形？

例题

例题3：【石子合并 ~~简化版~~】

解析：

- 回过神来！原题还没有解决！
- 最大得分？
  - 和最小得分相似，状态转移时min全部换成max即可
- n堆石子排成环形？
  - 处理环形问题，典型的办法时将其扩展为长度为2n的序列。
  - 将原来的n堆石子复制一遍接在后面，形成2n堆石子

例题

例题3：【石子合并 ~~简化版~~】

解析：

- 回过神来！原题还没有解决！
- 最大得分？
    - 和最小得分相似，状态转移时min全部换成max即可
- n堆石子排成环形？
    - 处理环形问题，典型的办法时将其扩展为长度为2n的序列。
    - 将原来的n堆石子复制一遍接在后面，形成2n堆石子
    - 执行动态规划算法，算出f[][]数组
    - 环形的最小得分 = min(f[1][n], f[2][n], ..., f[n-1][n])。
    - 时间复杂度仍然是$O(n^3)$。

例题

例题4：【NKOJ1507 做错的作业】

给出一个由大括号{ }、中括号[ ]和小括号( )构成的括号串，问最少添加几个括号，就能使得括号串中的括号成对匹配。

如下例所示：

([( ] { }) (< >))

只需添加两个括号即可，具体方法很多，例如这两种

方案1：(([( )] { }) (< >))

方案2：([ ] ([ ] { }) (< >))

例题

例题4：【NKOJ1507 做错的作业】

解析：

- 设定状态：类似合并石子，f[i][j]表示从第i个符号起的连续j个符号，最少要添几个符号

例题

例题4：【NKOJ1507 做错的作业】

解析：

- 设定状态：类似合并石子，f[i][j]表示从第i个符号起的连续j个符号，最少要添几个符号
- 边界情况：
  - j=0时，没有符号，也不需要添加符号，f[i][0] = 0
  - j=1时，只有一个符号，需要添加一个符号，f[i][1] = 1

例题

例题4：【NKOJ1507 做错的作业】

解析：

- 设定状态：类似合并石子，f[i][j]表示从第i个符号起的连续j个符号，最少要添几个符号
- 边界情况：
  - j=0时，没有符号，也不需要添加符号，f[i][0] = 0
  - j=1时，只有一个符号，需要添加一个符号，f[i][1] = 1
- 转移：考虑最后一个符号a[i+j-1]与谁配对
  - 不配对，f[i][j] ← f[i][j-1] + 1

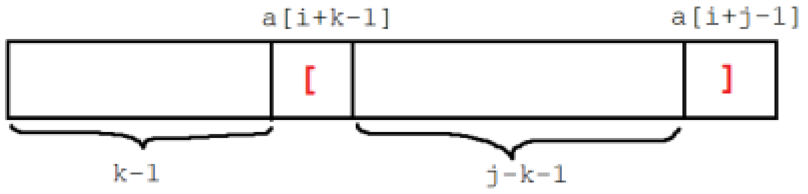
重庆南开中学

例题

例题4：【NKOJ1507 做错的作业】

解析：

- 设定状态：类似合并石子，f[i][j]表示从第i个符号起的连续j个符号，最少要添几个符号
- 边界情况：
  - j=0时，没有符号，也不需要添加符号，f[i][0] = 0
  - j=1时，只有一个符号，需要添加一个符号，f[i][1] = 1
- 转移：考虑最后一个符号a[i+j-1]与谁配对
  - 不配对，f[i][j] ← f[i][j-1] + 1
  - 与当前范围内的第k个符号a[i+k-1]配对，1≤k≤j-1
    - 需要a[i+k-1]和a[i+j-1]确实可以配对
    - f[i][j] ← f[i][k-1] + f[i+k+1][j-k-1]

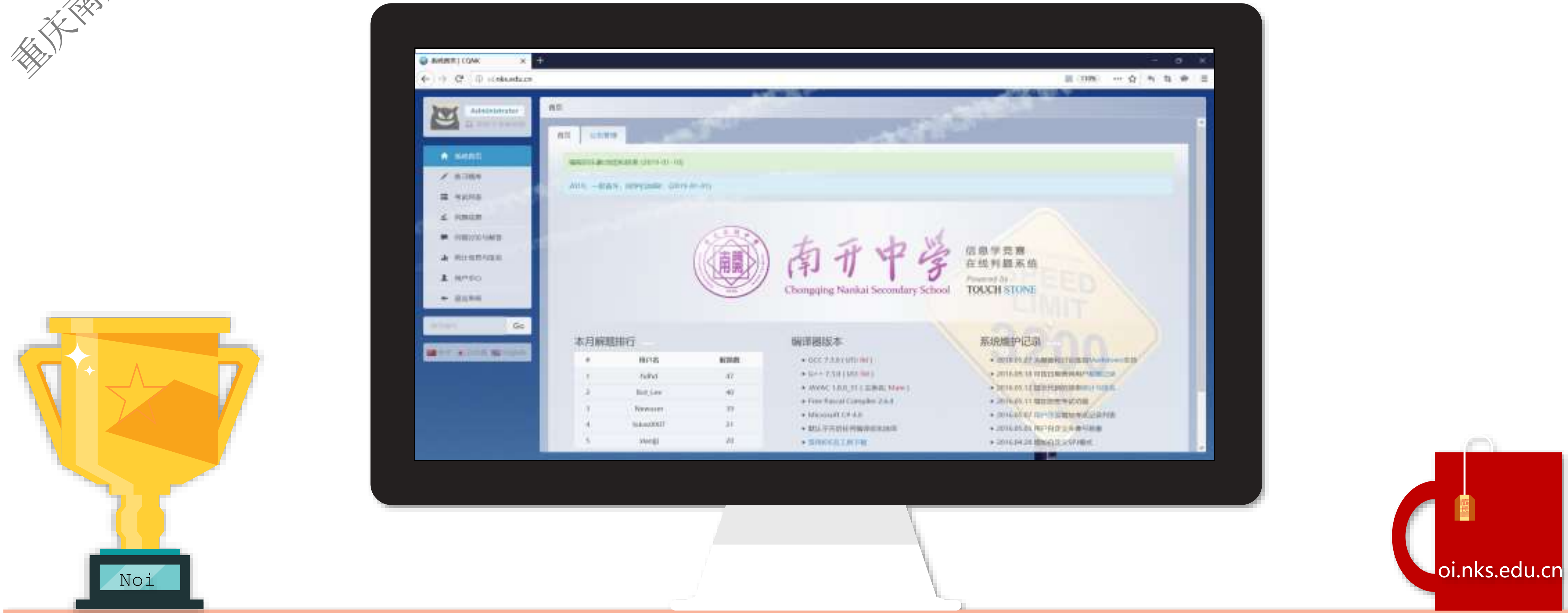


课后练习

动态规划习题

NKOJ1182	建别墅
NKOJ2553	开垦农田
NKOJ1068	最小交换合并
NKOJ1835	游戏
NKOJ1010	能量项链
NKOJ1507	做错的作业
NKOJ1036	回文词
NKOJ2008	涂色

重庆南开中学


南开中学
Chongqing Nankai Secondary School
信息学竞赛
在线判题系统
Powered by
TOUCH STONE
本月解题排行
编译器版本
系统维护记录
Noi
oi.nks.edu.cn