

最大子矩阵

<https://github.com/sangjianshun/Master-School/blob/master/getMaxMatrix.py>



问题描述

nums

0	-2	-7	0
9	2	-6	2
-4	1	-4	1
-1	7	0	-2



14

最大子矩阵

● 问题求解：动态规划

$$dp[i][j] = nums[0][j] + nums[1][j] + \dots + nums[i-1][j]$$

nums

0	-2	-7	0
9	2	-6	2
-4	1	-4	1
-1	7	0	-2

dp

0	0	0	0
0	-2	-7	0



0	0	0	0
0	-2	-7	0
9	0	-13	2



0	0	0	0
0	-2	-7	0
9	0	-13	2
5	1	-17	3
4	8	-17	1



0	0	0	0
0	-2	-7	0
9	0	-13	2
5	1	-17	3

最大子矩阵

● 问题求解：动态规划

nums

0	-2	-7	0
9	2	-6	2
-4	1	-4	1
-1	7	0	-2

dp

0	0	0	0
0	-2	-7	0
9	0	-13	2
5	1	-17	3
4	8	-17	1

辅助数组减少了大量的重复计算

dp[2]-dp[1]

9	2	-6	2
---	---	----	---

0	-2	-7	0
9	2	-6	2
-4	1	-4	1
-1	7	0	-2

9	11	5	7
---	----	---	---

dp[3]-dp[1]

5	3	-10	3
---	---	-----	---

0	-2	-7	0
9	2	-6	2
-4	1	-4	1
-1	7	0	-2

5	8	0	3
---	---	---	---

dp[4]-dp[1]

4	10	-10	1
---	----	-----	---

0	-2	-7	0
9	2	-6	2
-4	1	-4	1
-1	7	0	-2

4	14	4	5
---	----	---	---

```
for i in range(1,self.m+1):
    for j in range(i,self.m+1):
        tmp = 0
        for k in range(self.n):
            tmp+=(dp[j][k]-dp[i-1][k])
            if tmp<=0:
                tmp=0
            else:
                if tmp>res:
                    res = tmp
```