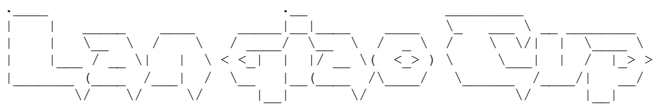
蓝桥杯2023



序言

此项目以刷题笔记为例，将所有markdown编写的md文档进行合并操作，最后转为pdf文档，简化了知识笔记产出的流程，可以极大满足汇总笔记的要求。

刷题链接：https://www.lanqiao.cn/cup/?sort=students_count&second_category_id=3

笔记列表：

省赛时间：2023-04-08(周六)

本项目`Git Https`链接：

> https://github.com/yaunsine/LanqiaoCup.git

本项目`Git SSH`链接:

> git@github.comryaunsine/LanqiaoCup.git

加载环境：

```
pip install -r requirements.txt
```

合并多个markdown

在`cmd`终端下运行

```
combine.bat
```

markdown文件转pdf方法：

```
python main.py
```

第三方工具

[1] `wkhtmltopdf`工具

[2] `grip`工具

1445.空间

本题为填空题，只需要算出结果后，在代码中使用输出语句将所填结果输出即可。

小蓝准备用 256MB 的内存空间开一个数组，数组的每个元素都是 32 位 二进制整数，如果不考虑程序占用的空间和维护内存需要的辅助空间，请问 256MB 的空间可以存储多少个 32 位二进制整数？

```
import os
import sys

# 请在此输入您的代码
print(256 * 1024 * 1024 * 8 // 32)
```

497.成绩分析

题目描述 小蓝给学生们组织了一场考试，卷面总分为 100 分，每个学生的得分都是一个 0 到 100 的整数。

请计算这次考试的最高分、最低分和平均分。

输入描述 输入的第一行包含一个整数 n (1≤n≤10 4)，表示考试人数。

接下来 n 行，每行包含一个 0 至 100 的整数，表示一个学生的得分。

输出描述 输出三行。

第一行包含一个整数，表示最高分。

第二行包含一个整数，表示最低分。

第三行包含一个实数，四舍五入保留正好两位小数，表示平均分。

输入输出样例 示例 输入

7 80 92 56 74 88 99 10 copy 输出

99 10 71.29

```
import os
import sys

# 请在此输入您的代码
n = int(input())  # 考试人数
sum_score = 0   # 总分
max_score = 0   # 最高分
min_score = 0   # 最低分
for i in range(n):
  score = int(input())  # 考试得分
  if i == 0:
    min_score = score
  if min_score > score:
    min_score = score
  if max_score < score:
    max_score = score
  sum_score += score

print(max_score)
print(min_score)
print("%.2f"%(sum_score / n))
```

此题注意细节

505数字三角形问题

上图给出了一个数字三角形。从三角形的顶部到底部有很多条不同的路径。对于每条路径，把路径上面的数加起来可以得到一个和，你的任务就是找到最大的和。

路径上的每一步只能从一个数走到下一层和它最近的左边的那个数或者右 边的那个数。此外，向左下走的次数与向右下走的次数相差不能超过 1。

输入描述 输入的第一行包含一个整数 N

(1 ≤ N ≤ 100) N (1≤N≤100)，表示三角形的行数。

下面的 N 行给出数字三角形。数字三角形上的数都是 0 至 100 之间的整数。

输出描述 输出一个整数，表示答案。

输入输出样例 示例 输入

```
5
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5
```

输出

27

代码实现：

```
import os
import sys

# 请在此输入您的代码
n = int(input())
i = 1
all_ls = list()
while i <= n:
  line = input()
  row_ls = line.split()
  row_ls = list(map(int, row_ls))
  all_ls.append(row_ls)
  i += 1


for i in range(1, n):
  for j in range(0, i + 1):
    if j == 0:
      all_ls[i][j] += all_ls[i - 1][j]
    elif j == i:
      all_ls[i][j] += all_ls[i - 1][j - 1]
    else:
      all_ls[i][j] += max(all_ls[i - 1][j - 1], all_ls[i - 1][j])

print(max(all_ls[-1][n // 2 - 1], all_ls[-1][n // 2]))
```

598.排序

题目描述 本题为填空题，只需要算出结果后，在代码中使用输出语句将所填结果输出即可。

小蓝最近学习了一些排序算法，其中冒泡排序让他印象深刻。

在冒泡排序中，每次只能交换相邻的两个元素。

小蓝发现，如果对一个字符串中的字符排序，只允许交换相邻的两个字符， 则在所有可能的排序方案中，冒泡排序的总交换次数是最少的。

例如，对于字符串 lan 排序，只需要 1 1 次交换。对于字符串 qiao 排序，总共需要 4 4 次交换。

小蓝找到了很多字符串试图排序，他恰巧碰到一个字符串，需要 100 100 次交 换，可是他忘了吧这个字符串记下来，现在找不到了。

请帮助小蓝找一个只包含小写英文字母且没有字母重复出现的字符串，对 该串的字符排序，正好需要 100 100 次交换。如果可能找到多个，请告诉小蓝最短的那个。如果最短的仍然有多个，请告诉小蓝字典序最小的那个。

```
import os
import sys

# 请在此输入您的代码
sum_x = 0
i = 1
```

```
for i in range(1, 101):
  sum_x += i
  if sum_x >= 100:
    break
x = sum_x - 100
result = str()
sum_x = 0
first_letter = str()
for j in range(i, -1, -1):
  if i - j == x:
    first_letter = str(chr(97+j))
  else:
    result += str(chr(97+j))
print(first_letter + result)
```

1463.货物摆放问题

题目描述 小蓝有一个超大的仓库，可以摆放很多货物。

现在，小蓝有 n 箱货物要摆放在仓库，每箱货物都是规则的正方体。小蓝规定了长、宽、高三个互相垂直的方向，每箱货物的边都必须严格平行于长、宽、高。

小蓝希望所有的货物最终摆成一个大的长方体。即在长、宽、高的方向上分别堆 L、 W、 H 的货物,满足 n=L×W×H。

给定 n，请问有多少种堆放货物的方案满足要求。

例如，当 n=4 时，有以下 6 6 种方案： 1 × 1 × 4 、 1 × 2 × 2 、 1 × 4 × 1 、 2 × 1 × 2 、 2 × 2 × 1 、 4 × 1 × 1 1×1×4、1×2×2、1×4×1、2×1×2、2×2×1、4×1×1。

请问，当 n=2021041820210418（注意有 16 16 位数字）时，总共有多少种方案？

提示：建议使用计算机编程解决问题。

答案提交 这是一道结果填空的题，你只需要算出结果后提交即可。本题的结果为一个整数，在提交答案时只填写这个整数，填写多余的内容将无法得分。

运行限制 最大运行时间：1s 最大运行内存: 256M

【解法1】求因子列表，重新组合（需要在pycharm运行才能不超时）

```
import os
import sys

# 请在此输入您的代码
# n = L * W * H 因式分解*3

"""
import os
import sys
# 请在此输入您的代码
# n = L * W * H 因式分解*3

ans = 0
n = 2021041820210418
factor_list = []
j = 0
ans = 0
for i in range(1, int(pow(n, 0.5))):
    if n % i == 0:
        factor_list.append(i)
        j = n // i  # n = i * j = i * (x * y) = (x * y) * j
        if j != i:
            factor_list.append(j)

print(len(factor_list))

for i in factor_list:
    for j in factor_list:
        for k in factor_list:
            if i * j * k == n:
                ans += 1

print(ans)
"""
print(2430)
```

【解法2】 大佬的解法，可以直接在蓝桥云平台运行

```
import os
import sys
# 因素+背包（求因子）+遍历
# 不到50毫秒！！！！！！！！！！
# 请在此输入您的代码
n=2021041820210418
l=[]     # ！！！！用于存因数不是因子例如：10=2*5
i=2
x=n
while i<pow(x, 0.5):
    if x%i==0:
        l.append(i)
        x=x//i
    else:
        i+=1

l.append(x)
s={1}   # ！！！！用于存因子 如10=1*2*5*10
for j in l:
    s = set([j*k for k in s]) | s
count=0
for k1 in s:             # 遍历两层求解
    for k2 in s:
        if n%(k1*k2)==0:
            count+=1
print(count)
```

819.递增序列

【解法1】 暴力超时解法：

```
import os
import sys

# 请在此输入您的代码
# 双指针解法(i, j)->(m, n)递增
# (i, j) - (i, n)
# (i, j) - (m, j)
# (i, j) - (i + x, j + x)
text = """VLPWJVVNNZSWFGHSFRBCOIJTPYNEURPIGKQGPSXUGNELGRVZAG
SDLLOVGRTWEYZKKXNKIRWGZWXWRHKXFASATDWZAPZRNHTNNGQF
```

```
ZGUGXVQDQAEAHOQEADMWWXFBXECKAVIGPTKTTQFWSWPKRPSMGA
BDGMGYHAOPPRRHKYZCMFZEDELCALTBSWNTAODXYVHQNDASUFRL
YVYWQZUTEPFSFXLTZBMBQETXGXFUEBHGMJKBPNIHMYOELYZIKH
ZYZHSLTCGNANNXTUJGBYKUOJMGOGRDPKEUGVHNZJZHDUNRERBU
XFPTZKTPVQPJEMBHNTUBSMIYEGXNWQSBZMHMDRZZMJPZQTCWLR
ZNXOKBITTPSHEXWHZXFLWEMPZTBVNKNYSHCIQRIKQHFRAYWOPG
MHJKFYYBQSDPOVJICWWGGCOZSBGLSOXOFDAADZYEOBKDDTMQPA
VIDPIGELBYMEVQLASLQRUKMXSEWGHRSFVXOMHSJWWXHIBCGVIF
GWRFRFLHAMYWYZOIQODBIHHRIIMWJWJGYPFAHZZWJKRGOISUJC
EKQKKPNEYCBWOQHTYFHHQZRLFNDOVXTWASSQWXKBIVTKTUIASK
PEKNJFIVBKOZUEPPHIWLUBFUDWPIDRJKAZVJKPBRHCRMGNMFWW
CGZAXHXPDELTACGUWBXWNNZNDQYYCIQRJCULIEBQBLLMJEUSZP
RWHHQMBIJWTQPUFNAESPZHAQARNIDUCRYQAZMNVRVZUJOZUDGS
PFGAYBDEECHUXFUZIKAXYDFWJNSAOPJYWUIEJSCORRBVQHCHMR
JNVIPVEMQSHCCAXMWEFSYIGFPIXNIDXOTXTNBCHSHUZGKXFECL
YZBAIIOTWLREPZISBGJLQDALKZUKEQMKLDIPXJEPENEIPWFDLP
HBQKWJFLSEXVILKYPNSWUZLDCRTAYUUPEITQJEITZRQMMAQNLN
DQDJGOWMBFKAIGWEAJOISPFPLULIWVVALLIIHBGEZLGRHRCKGF
LXYPCVPNUKSWCCGXEYTEBAWRLWDWNHHNNNWQNIIBUCGUJYMRYW
CZDKISKUSBPFHVGSAVJBDMNPSDKFRXVVPLVAQUGVUJEXSZFGFQ
IYIJGISUANRAXTGQLAVFMQTICKQAHLEBGHAVOVVPEXIMLFWIYI
ZIIFSOPCMAWCBPKWZBUQPQLGSNIBFADUUJJHPAIUVVNWNWKDZB
HGTEEIISFGIUEUOWXVTPJDVACYQYFQUCXOXOSSMXLZDQESHXKP
FEBZHJAGIFGXSMRDKGONGELOALLSYDVILRWAPXXBPOOSWZNEAS
VJGMAOFLGYIFLJTEKDNIWHJAABCASFMAKIENSYIZZSLRSUIPCJ
BMQGMPDRCPGWKTPLOTAINXZAAJWCPUJHPOUYWNWHZAKCDMZDSR
RRARTVHZYYCEDXJQNQAINQVDJCZCZLCQWQQIKUYMYMOVMNCBVY
ABTCRRUXVGYLZILFLOFYVWFFBZNFWDZOADRDCLIRFKBFBHMAXX"""
mat = [list(line) for line in text.split("\n")]
ans = 0
rows = len(mat)
cols = len(mat[0])
for i in range(rows):
  for j in range(cols):
    for m in range(rows):
      for n in range(cols):
        if (j < n and mat[i][j] < mat[i][n]) or (i < m and mat[i][j] < mat[m][j]) or (not(m <= i and n <= j) and abs(m - i) == abs(n - j) and mat[i][j] < mat[m][n]):
          ans += 1
print(ans)
```

【解法2】 正确解法:

```
import os
import sys

# 请在此输入您的代码
# 双指针解法(i, j)->(m, n)递增
# (i, j) - (i, n)
# (i, j) - (m, j)
# (i, j) - (i + x, j + x)
text = """VLPWJVVNNZSWFGHSFRBCOIJTPYNEURPIGKQGPSXUGNELGRVZAG
SDLLOVGRTWEYZKKXNKIRWGZWXWRHKXFASATDWZAPZRNHTNNGQF
ZGUGXVQDQAEAHOQEADMWWXFBXECKAVIGPTKTTQFWSWPKRPSMGA
BDGMGYHAOPPRRHKYZCMFZEDELCALTBSWNTAODXYVHQNDASUFRL
YVYWQZUTEPFSFXLTZBMBQETXGXFUEBHGMJKBPNIHMYOELYZIKH
ZYZHSLTCGNANNXTUJGBYKUOJMGOGRDPKEUGVHNZJZHDUNRERBU
XFPTZKTPVQPJEMBHNTUBSMIYEGXNWQSBZMHMDRZZMJPZQTCWLR
ZNXOKBITTPSHEXWHZXFLWEMPZTBVNKNYSHCIQRIKQHFRAYWOPG
MHJKFYYBQSDPOVJICWWGGCOZSBGLSOXOFDAADZYEOBKDDTMQPA
VIDPIGELBYMEVQLASLQRUKMXSEWGHRSFVXOMHSJWWXHIBCGVIF
GWRFRFLHAMYWYZOIQODBIHHRIIMWJWJGYPFAHZZWJKRGOISUJC
EKQKKPNEYCBWOQHTYFHHQZRLFNDOVXTWASSQWXKBIVTKTUIASK
PEKNJFIVBKOZUEPPHIWLUBFUDWPIDRJKAZVJKPBRHCRMGNMFWW
CGZAXHXPDELTACGUWBXWNNZNDQYYCIQRJCULIEBQBLLMJEUSZP
RWHHQMBIJWTQPUFNAESPZHAQARNIDUCRYQAZMNVRVZUJOZUDGS
PFGAYBDEECHUXFUZIKAXYDFWJNSAOPJYWUIEJSCORRBVQHCHMR
JNVIPVEMQSHCCAXMWEFSYIGFPIXNIDXOTXTNBCHSHUZGKXFECL
YZBAIIOTWLREPZISBGJLQDALKZUKEQMKLDIPXJEPENEIPWFDLP
HBQKWJFLSEXVILKYPNSWUZLDCRTAYUUPEITQJEITZRQMMAQNLN
DQDJGOWMBFKAIGWEAJOISPFPLULIWVVALLIIHBGEZLGRHRCKGF
LXYPCVPNUKSWCCGXEYTEBAWRLWDWNHHNNNWQNIIBUCGUJYMRYW
CZDKISKUSBPFHVGSAVJBDMNPSDKFRXVVPLVAQUGVUJEXSZFGFQ
IYIJGISUANRAXTGQLAVFMQTICKQAHLEBGHAVOVVPEXIMLFWIYI
ZIIFSOPCMAWCBPKWZBUQPQLGSNIBFADUUJJHPAIUVVNWNWKDZB
HGTEEIISFGIUEUOWXVTPJDVACYQYFQUCXOXOSSMXLZDQESHXKP
FEBZHJAGIFGXSMRDKGONGELOALLSYDVILRWAPXXBPOOSWZNEAS
VJGMAOFLGYIFLJTEKDNIWHJAABCASFMAKIENSYIZZSLRSUIPCJ
BMQGMPDRCPGWKTPLOTAINXZAAJWCPUJHPOUYWNWHZAKCDMZDSR
RRARTVHZYYCEDXJQNQAINQVDJCZCZLCQWQQIKUYMYMOVMNCBVY
ABTCRRUXVGYLZILFLOFYVWFFBZNFWDZOADRDCLIRFKBFBHMAXX"""
mat = [list(line) for line in text.split("\n")]
ans = 0
rows = len(mat)
cols = len(mat[0])
for i in range(rows):
  for j in range(cols):
    for m in range(rows):
      for n in range(cols):
        if mat[i][j] < mat[m][n] and ((i == m and j < n) or (j == n and i < m) or (abs(m - i) == abs(n - j) and not (m <= i and n <= j))):
          ans += 1
print(ans)
```

【解法2-2】 拆分复杂`if`

```
import os
import sys

# 请在此输入您的代码
# 双指针解法(i, j)->(m, n)递增
# (i, j) - (i, n)
# (i, j) - (m, j)
# (i, j) - (i + x, j + x)
text = """VLPWJVVNNZSWFGHSFRBCOIJTPYNEURPIGKQGPSXUGNELGRVZAG
SDLLOVGRTWEYZKKXNKIRWGZWXWRHKXFASATDWZAPZRNHTNNGQF
ZGUGXVQDQAEAHOQEADMWWXFBXECKAVIGPTKTTQFWSWPKRPSMGA
BDGMGYHAOPPRRHKYZCMFZEDELCALTBSWNTAODXYVHQNDASUFRL
YVYWQZUTEPFSFXLTZBMBQETXGXFUEBHGMJKBPNIHMYOELYZIKH
ZYZHSLTCGNANNXTUJGBYKUOJMGOGRDPKEUGVHNZJZHDUNRERBU
XFPTZKTPVQPJEMBHNTUBSMIYEGXNWQSBZMHMDRZZMJPZQTCWLR
ZNXOKBITTPSHEXWHZXFLWEMPZTBVNKNYSHCIQRIKQHFRAYWOPG
MHJKFYYBQSDPOVJICWWGGCOZSBGLSOXOFDAADZYEOBKDDTMQPA
VIDPIGELBYMEVQLASLQRUKMXSEWGHRSFVXOMHSJWWXHIBCGVIF
GWRFRFLHAMYWYZOIQODBIHHRIIMWJWJGYPFAHZZWJKRGOISUJC
EKQKKPNEYCBWOQHTYFHHQZRLFNDOVXTWASSQWXKBIVTKTUIASK
PEKNJFIVBKOZUEPPHIWLUBFUDWPIDRJKAZVJKPBRHCRMGNMFWW
CGZAXHXPDELTACGUWBXWNNZNDQYYCIQRJCULIEBQBLLMJEUSZP
RWHHQMBIJWTQPUFNAESPZHAQARNIDUCRYQAZMNVRVZUJOZUDGS
PFGAYBDEECHUXFUZIKAXYDFWJNSAOPJYWUIEJSCORRBVQHCHMR
JNVIPVEMQSHCCAXMWEFSYIGFPIXNIDXOTXTNBCHSHUZGKXFECL
YZBAIIOTWLREPZISBGJLQDALKZUKEQMKLDIPXJEPENEIPWFDLP
```

```
HBQKWJFLSEXVILKYPNSWUZLDCRTAYUUPEITQJEITZRQMMAQNLN
DQDJGOWMBFKAIGWEAJOISPFPLULIWVVALLIIHBGEZLGRHRCKGF
LXYPCVPNUKSWCCGXEYTEBAWRLWDWNHHNNNWQNIIBUCGUJYMRYW
CZDKISKUSBPFHVGSAVJBDMNPSDKFRXVVPLVAQUGVUJEXSZFGFQ
IYIJGISUANRAXTGQLAVFMQTICKQAHLEBGHAVOVVPEXIMLFWIYI
ZIIFSOPCMAWCBPKWZBUQPQLGSNIBFADUUJJHPAIUVVNWNWKDZB
HGTEEIISFGIUEUOWXVTPJDVACYQYFQUCXOXOSSMXLZDQESHXKP
FEBZHJAGIFGXSMRDKGONGELOALLSYDVILRWAPXXBPOOSWZNEAS
VJGMAOFLGYIFLJTEKDNIWHJAABCASFMAKIENSYIZZSLRSUIPCJ
BMQGMPDRCPGWKTPLOTAINXZAAJWCPUJHPOUYWNWHZAKCDMZDSR
RRARTVHZYYCEDXJQNQAINQVDJCZCZLCQWQQIKUYMYMOVMNCBVY
ABTCRRUXVGYLZILFLOFYVWFFBZNFWDZOADRDCLIRFKBFBHMAXX"""
mat = [list(line) for line in text.split("\n")]
ans = 0
rows = len(mat)
cols = len(mat[0])
for i in range(rows):
  for j in range(cols):
    for m in range(rows):
      for n in range(cols):
        if mat[i][j] < mat[m][n]:
          if i == m and j < n:
            ans += 1
          if j == n and i < m:
            ans += 1
          if abs(m - i) == abs(n - j) and not (m <= i and n <= j):
            ans += 1
print(ans)
```

597.跑步训练

【超时解法】

```
import os
import sys

# 判断是否闰年
def judge_leap_year(year) -> bool:
  if year % 400 == 0:
    return True
  if year % 100 != 0 and year % 4 == 0:
    return True
  return False

# 计算每年的天数
def calculate_days(year) -> int:
  if judge_leap_year(year):
    return 366
  else:
    return 365

# 计算每月的天数
def month_days(month, year=2020) -> int:
  big_m = [1, 3, 5, 7, 8, 10, 12]
  small_m = [4, 6, 9, 11]
  if month == 2:
    return 29 if judge_leap_year(year) else 28
  elif month in big_m:
    return 31
  elif month in small_m:
    return 30

y = 2000
m = 1
d = 1
wd = 6
days_metres = 0

while y < 2020:
  if wd == 1 and d == 1:
    days_metres += 2
  elif wd == 1:
    days_metres += 2
  elif d == 1:
    days_metres += 2
  else:
    days_metres += 1

  d += 1
  wd += 1
  if wd > 7:
    wd = 1
  if month_days(m) > d:
    m += 1
    d = 1
  if m > 12:
    m = 1
    y += 1

while m < 10:
  if wd == 1 and d == 1:
    days_metres += 2
  elif wd == 1:
    days_metres += 2
  elif d == 1:
    days_metres += 2
  else:
    days_metres += 1

  d += 1
  wd += 1
  if wd > 7:
    wd = 1
  if month_days(m) > d:
    m += 1
    d = 1
  if m > 12:
    m = 1
    y += 1

while d <= 1:
  if wd == 1 and d == 1:
    days_metres += 2
  elif wd == 1:
    days_metres += 2
  elif d == 1:
    days_metres += 2
  else:
    days_metres += 1
```

```
  d += 1
  wd += 1
  if wd > 7:
    wd = 1
  if month_days(m) > d:
    m += 1
    d = 1
  if m > 12:
    m = 1
    y += 1
print(days_metres)
```

【算法2】

借助datetime库

```
import os
import sys

import datetime

start = datetime.date(2000, 1, 1)
end = datetime.date(2020, 10, 1)
days = datetime.timedelta(days=1)
ans = 0

while start <= end:
  if start.day == 1 or start.weekday() == 0:
    ans += 2
  else:
    ans += 1
  start += days

print(ans)
```

【解法3】

暴力求解

```
import os
import sys

def get_kilometres():
  months = [31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

  cnt = 0
  ans = 6
  for year in range(2000, 2021, 1):
    # 判断闰年
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
      months[1] = 29
    else:
      months[1] = 28
    for i, m in enumerate(months):
      for d in range(1, m + 1, 1):
        cnt += 1
        if ans > 7:
          ans = 1
        if ans == 1 or d == 1:
          cnt += 1
        ans += 1  # 星期更新
        if year == 2020 and i + 1 == 10 and d == 1:
          return cnt
print(get_kilometres())
```

594.蛇形填数

题目描述 本题为填空题，只需要算出结果后，在代码中使用输出语句将所填结果输出即可。

如下图所示，小明用从 1 开始的正整数“蛇形”填充无限大的矩阵。

```
1 2 6 7 15 ...
3 5 8 14 ...
4 9 13 ...
10 12 ...
11 ...
...
```

容易看出矩阵第二行第二列中的数是 5。请你计算矩阵中第 20 行第 20 列的数是多少？

【解法1】

模拟动态规划

```
import os
import sys

mat = [[0 for _ in range(200)] for _ in range(200)]

i = 0
j = 0
mat[i][j] = 1
cnt = 1
while mat[19][19] == 0:
  # 右移
  j += 1
  cnt += 1
  mat[i][j] =  cnt
  # 左下角
  while j != 0:
    i += 1
    j -= 1
    cnt += 1
    mat[i][j] = mat[i][j]
  # 下移
  i += 1
  cnt += 1
  mat[i][j] = cnt
  # 右上角
  while i != 0:
    i -= 1
    j += 1
    cnt += 1
    mat[i][j] = cnt

print(mat[19][19])
```

【解法2】

对角线计算法

```
res = 1
for i in range(0, 20, 1):
    res += i * 4
print(res)
```