# [279. 完全平方数](https://leetcode-cn.com/problems/perfect-squares/)

给定正整数 *n*，找到若干个完全平方数（比如 1, 4, 9, 16, ...）使得它们的和等于*n*。你需要让组成和的完全平方数的个数最少。

**示例 1:**

**输入:** *n* = 12

**输出:** 3

**解释:** 12 = 4 + 4 + 4.

**示例 2:**

**输入:** *n* = 13

**输出:** 2

**解释:** 13 = 4 + 9.

思路：

visited=[False(n)]:一组长度为n的False数组，用来记录visited[i]是否已经被访问。因为后面访问到相同数值的话，step的长度肯定是大于之前的step。所以被访问过一个数值i就把visited[i]置为True。

res[[n,step]]：记录数值和其经过的step；

每次pop出res第一个数值，循环遍历其完全平方数。

class Solution(object):

def numSquares(self, n):

"""

:type n: int

:rtype: int

"""

res = [];

res.append([n,0]);

visited = [False for \_ in range(n+1)];

visited[n] = True;

while res:

tmp,step = res.pop(0);

i = 1;

muti = tmp - i\*i;

while muti >= 0:

if muti == 0:

return step + 1;

if not visited[muti]:

res.append([muti,step+1]);

visited[muti] = True;

i += 1;

muti = tmp - i\*i;