Problem # 735 : Asteroid Collision

<https://leetcode.com/problems/asteroid-collision>

My Solution:

<https://leetcode.com/problems/asteroid-collision/discuss/905396/Simple-Python-3-Solution-Runtime-beats-91.23>

Simple Python 3 Solution -- Runtime beats 91.23%

1. Use the 'Last In First Out' data structure stack to push asteroids from the list into it.
2. while the stack is not empty, compare the top 2 elements in the stack. If the top element is negative(moving left) and the second element from the top is positive (moving right), they will collide.  
   Note: if the top two elements are of the same sign, they will not collide.  
   Again if the top element is positive(moving right) and the second element from the top is negative (moving left), they are going in opposite directions and so they will not collide.
3. If two elements collide in step 2, then the one with bigger size (i.e. absolute value) destroys the other. However, if they are of the same size, both are destroyed in the collision.
4. Return the resulting stack.

class Solution:

def asteroidCollision(self, asteroids: List[int]) -> List[int]:

stack = []

for asteroid in asteroids:

stack.append(asteroid)

while len(stack) > 1 and stack[-1] < 0 and stack[-2] > 0:

if stack[-2] > abs(stack[-1]): # top asteroid in the stack is smaller in size

stack.pop() # top asteroid in the stack is destroyed

elif stack[-2] < abs(stack[-1]): # second asteroid from top of the stack is smaller in size

stack.pop(-2) # second asteroid from top is destroyed

else: # both are equal in size and so pop both as both are destroyed

stack.pop()

stack.pop()

return(stack)