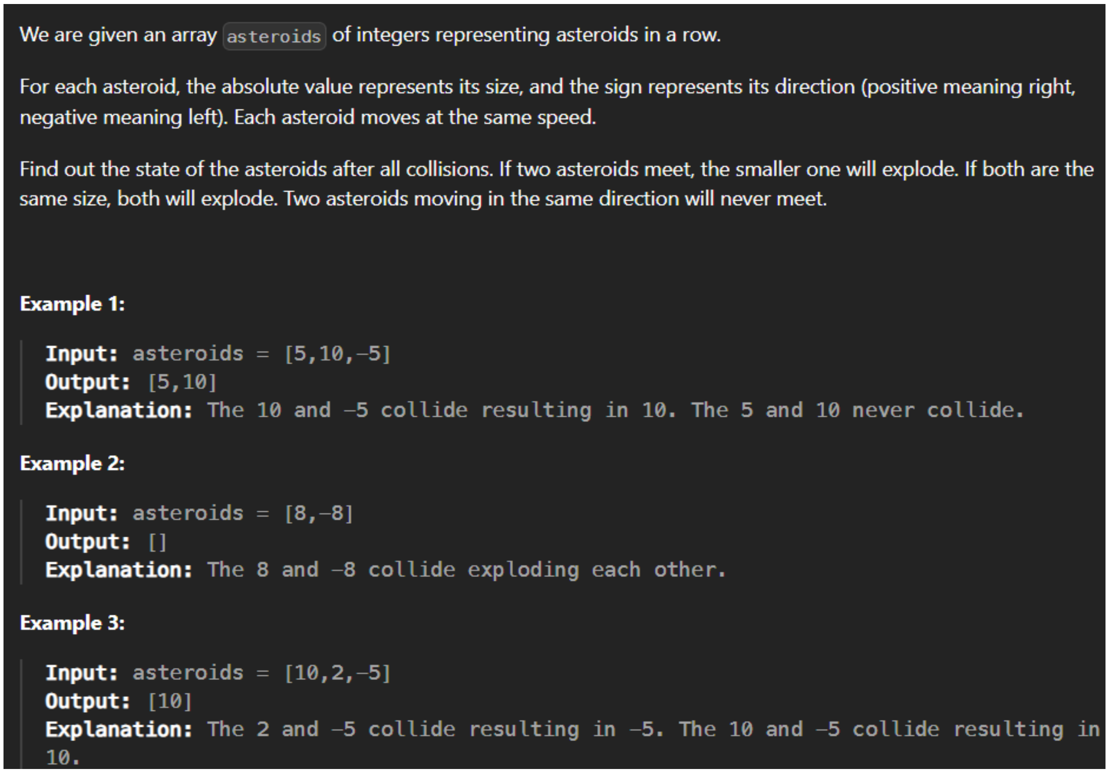
735. Asteroid Collision

Problem Statement

1. 

We are given an array `asteroids` of integers representing asteroids in a row.

For each asteroid, the absolute value represents its size, and the sign represents its direction (positive meaning right, negative meaning left). Each asteroid moves at the same speed.

Find out the state of the asteroids after all collisions. If two asteroids meet, the smaller one will explode. If both are the same size, both will explode. Two asteroids moving in the same direction will never meet.

Example 1:

```
Input: asteroids = [5,10,-5]
Output: [5,10]
Explanation: The 10 and -5 collide resulting in 10. The 5 and 10 never collide.
```

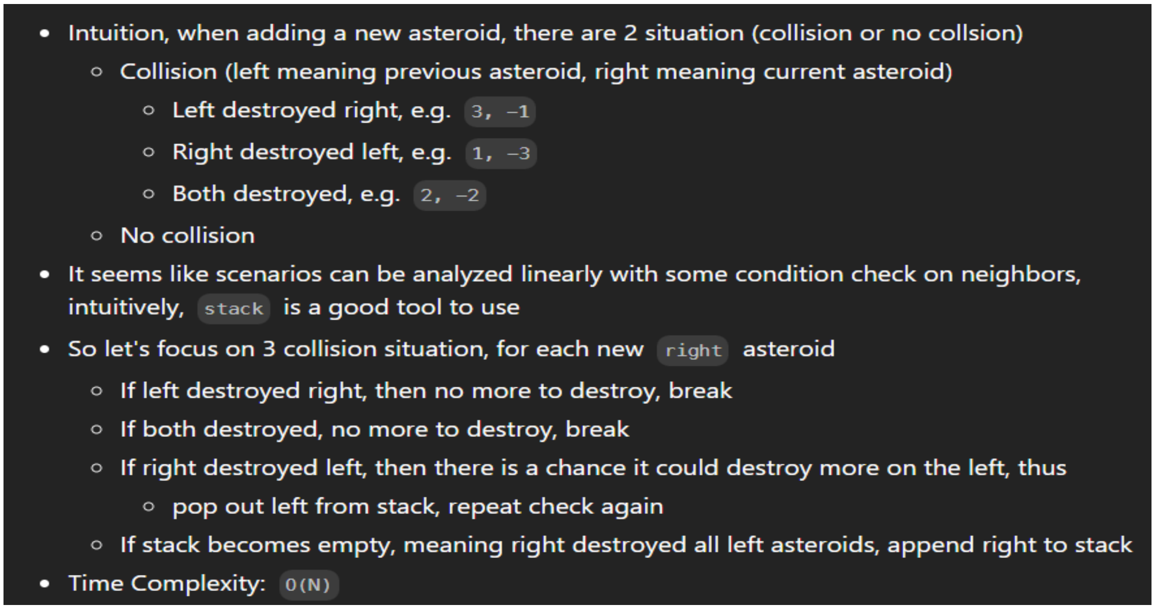
Example 2:

```
Input: asteroids = [8,-8]
Output: []
Explanation: The 8 and -8 collide exploding each other.
```

Example 3:

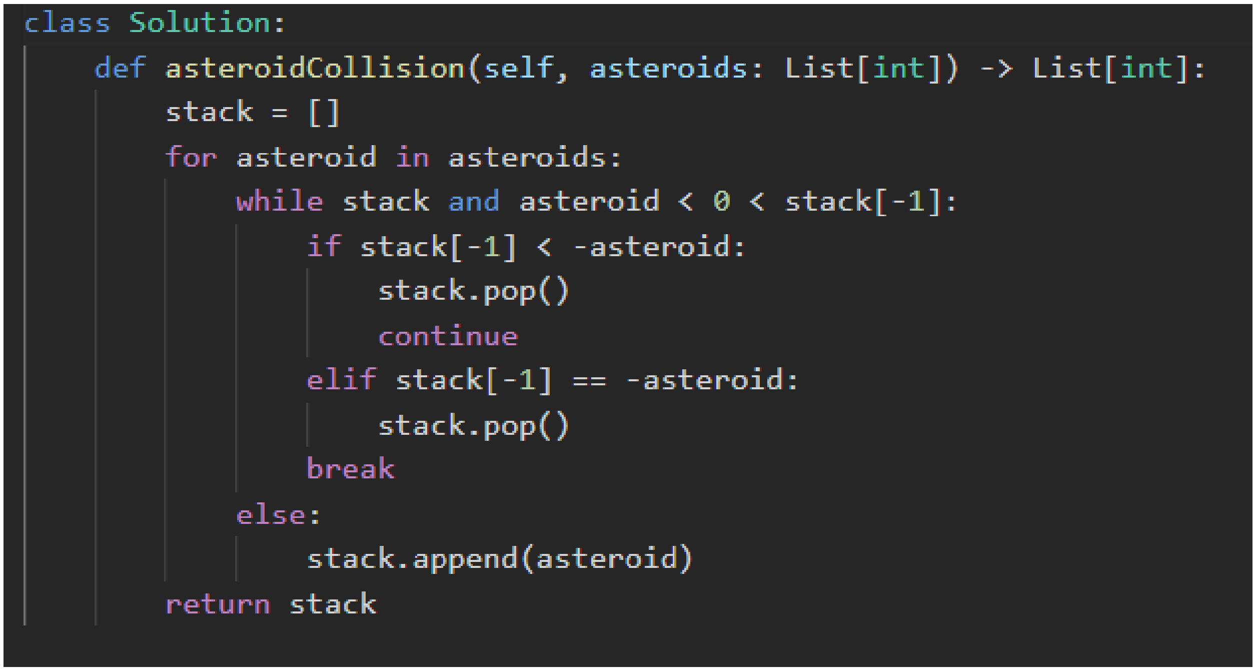
```
Input: asteroids = [10,2,-5]
Output: [10]
Explanation: The 2 and -5 collide resulting in -5. The 10 and -5 collide resulting in
10.
```

Approach

1. 

- Intuition, when adding a new asteroid, there are 2 situation (collision or no collsion)
  - Collision (left meaning previous asteroid, right meaning current asteroid)
    - Left destroyed right, e.g. `3, -1`
    - Right destroyed left, e.g. `1, -3`
    - Both destroyed, e.g. `2, -2`
  - No collision
- It seems like scenarios can be analyzed linearly with some condition check on neighbors, intuitively, `stack` is a good tool to use
- So let's focus on 3 collision situation, for each new `right` asteroid
  - If left destroyed right, then no more to destroy, break
  - If both destroyed, no more to destroy, break
  - If right destroyed left, then there is a chance it could destroy more on the left, thus
    - pop out left from stack, repeat check again
  - If stack becomes empty, meaning right destroyed all left asteroids, append right to stack
- Time Complexity: `O(N)`

Python code:

1. 

```python
class Solution:
    def asteroidCollision(self, asteroids: List[int]) -> List[int]:
        stack = []
        for asteroid in asteroids:
            while stack and asteroid < 0 < stack[-1]:
                if stack[-1] < -asteroid:
                    stack.pop()
                    continue
                elif stack[-1] == -asteroid:
                    stack.pop()
                break
            else:
                stack.append(asteroid)
        return stack
```