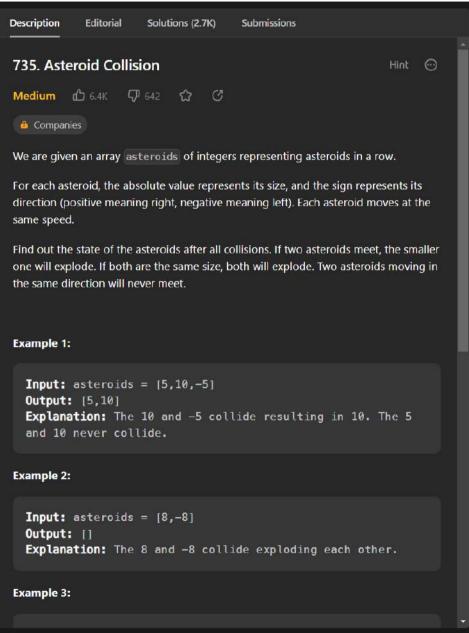


```
i Java · Auto
         public int search(int[] nums, int target) {
             int start = 0, end = nums.length - 1;
             int mid = (start + end) / 2;
             while (start <= end) {
                 mid = (start + end) / 2;
                 if (target == nums[mid]) {
                      return mid;
                 if (nums[start] <= nums[mid]) {
                      if (nums[start] <= target && nums[mid] >= target) {
                          end = mid - 1;
                      } else {
                          start = mid + 1;
                  } else {
                      if (nums[end] >= target && nums[mid] <= target) {</pre>
                          start = mid + 1.
Testcase
Accepted Runtime: 0 ms
  • Case 1
                          • Case 3
              • Case 2
Input
  [4,5,6,7,0,1,2]
 0
Output
Console v
                                                                                  Run
                                                                                           Submit
```



```
i Java · Auto
    import java.util.*;
         public int[] asteroidCollision(int[] asteroids) {
             int n = asteroids.length;
             Deque<Integer> stk = new ArrayDeque<>();
             for (int i = 0; i < n; i++) {
                 if (stk.isEmpty() || asteroids[i] > 0) {
                     stk.push(asteroids[i]);
                 } else {
                     while (!stk.isEmpty() && stk.peek() > 0 && stk.peek() < Math.abs(asteroids
     [i])) {
                         stk.pop();
                      if (!stk.isEmpty() && stk.peek() == Math.abs(asteroids[i])) {
Testcase
       Result
Accepted Runtime: 0 ms
  Case 1
              Case 2
                          Case 3
Input
  [5, 10, -5]
Output
  [5,10]
Eupartad
Console v
                                                                                Run
                                                                                          Submit
```