JS

function make\_squares(arr) {

const n = arr.length;

squares = Array(n).fill(0);

let highestSquareIdx = n - 1;

let left = 0,

right = n - 1;

while (left <= right) {

let leftSquare = arr[left] \* arr[left],

rightSquare = arr[right] \* arr[right];

if (leftSquare > rightSquare) {

squares[highestSquareIdx] = leftSquare;

left += 1;

} else {

squares[highestSquareIdx] = rightSquare;

right -= 1;

}

highestSquareIdx -= 1;

}

return squares;

}

console.log(`Squares: ${make\_squares([-2, -1, 0, 2, 3])}`);

console.log(`Squares: ${make\_squares([-3, -1, 0, 1, 2])}`);

JAVA

class SortedArraySquares {

public static int[] makeSquares(int[] arr) {

int n = arr.length;

int[] squares = new int[n];

int highestSquareIdx = n - 1;

int left = 0, right = arr.length - 1;

while (left <= right) {

int leftSquare = arr[left] \* arr[left];

int rightSquare = arr[right] \* arr[right];

if (leftSquare > rightSquare) {

squares[highestSquareIdx--] = leftSquare;

left++;

} else {

squares[highestSquareIdx--] = rightSquare;

right--;

}

}

return squares;

}

public static void main(String[] args) {

int[] result = SortedArraySquares.makeSquares(new int[] { -2, -1, 0, 2, 3 });

for (int num : result)

System.out.print(num + " ");

System.out.println();

result = SortedArraySquares.makeSquares(new int[] { -3, -1, 0, 1, 2 });

for (int num : result)

System.out.print(num + " ");

System.out.println();

}

}