[Squares of Sorted Array](https://leetcode.com/problems/squares-of-a-sorted-array/)

**package** unsolvedpackage;

**public** **class** SquareOfSortedArray {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int**[] A = {-4,-1,0,3,10};

**int**[] result = *sortedSquares*(A);

**for**(**int** i : result)

System.***out***.print(i +" ");

}

**public** **static** **int**[] sortedSquares(**int**[] A) {

//base case

**if**(A == **null** || A.length == 0)

**return** **null**;

//sorted result array

**int**[] result = **new** **int**[A.length];

**int** low = 0;

**int** high = A.length - 1;

**int** index = A.length - 1;

**while**(index >= 0) {

//check if the absolute value of left element is greater than right most element, then swap

**if**(Math.*abs*(A[low]) >= Math.*abs*(A[high])) {

result[index] = A[low] \* A[low];

low++;

}

**else** {

result[index] = A[high] \* A[high];

high--;

}

index--;

}

**return** result;

}

}

Time Complexity : O(logn) , n is number of elements in array A.

Space Complexity : O(n), n is number of elements in array A.