GOLIBRARY

(https://www.golibrary.co/)

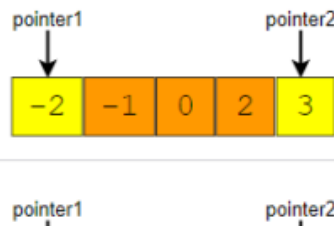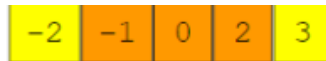## SQUARING A SORTED ARRAY

Home (https://www.golibrary.co)   >

Algorithms (https://www.golibrary.co/category/algorithms/)   >

javascript (https://www.golibrary.co/category/programming/javascript/)   >

Programming (https://www.golibrary.co/category/programming/)   >

Squaring a sorted array

💬 0 / 👁 288

SEARCH

Algorithm to find all factors of very large number (https://www.golibrary.co/algorithm-to-find-all-factors-of-very-large-number/)

Introduction and overview   Given a very large number, find all it's factors using an efficient algorithm. ... Read more... (https://www.golibrary.co/algorithm-to-find-all-factors-of-very-large-number/)

Algorithm to find prime factors of very big numbers (https://www.golibrary.co/algorithm-to-find-prime-factors-of-very-big-numbers/)

Problem Statement

Given a sorted array, create a new array containing **squares of all the number of the input array** in the sorted order.

Example 1:

Input: [-2, −1, 0, 2, 3]
Output: [0, 1, 4, 4, 9]

Example 2:

Input: [-3, −1, 0, 1, 2]

Output: [0 1 1 4 9]

Solution 1

One simple solution is to loop through the array elements and square them in place and then store the sorted squared array in output squares array and return it.

Javascript code below:-

```
 1  <script>
 2  function make_squares(arr) {
 3    squares = []
 4    // TODO: Write your code here
 5    arr = arr.map(item=>item*item);
 6    squares = arr.sort((a,b)=>a-b);
 7    return squares;
 8  };
 9
10  document.write(`Squares: ${make_squares([-2, -1, 0, 2, 3])}<br/>`);
11  document.write(`Squares: ${make_squares([-3, -1, 0, 1, 2])}<br/>`);
12  </script>
13
```

Time complexity

The time complexity of the above algorithm will be **O(N log N)** as we are first sorting the array and then iterating the input array only once.

Space complexity

The space complexity of the above algorithm will also be **O(N);** this space will be used for the output array.

The above solution works but isn't efficient as we are sorting and then iterating through the array again to store it in squares array. Better solution would be to use a 2 pointer approach and binary search mechanism to insert the squared elements in a sorted way as we iterate through the array.

Solution 2

This question becomes tricky because the array has negative numbers as well. An easier approach could be to first find the index of the first non-negative number in the array. After that, we can use **Two Pointers** to iterate the array. One pointer will move forward to iterate the non-negative numbers and the other pointer will move backward to iterate the negative numbers. At any step, whichever number gives us a bigger square will be added to the output array.

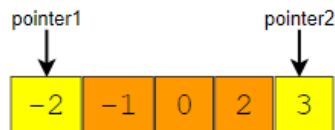For the above-mentioned Example-1, we will do something like this:
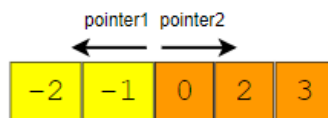
Squared Sorted Array1

Since the numbers at both the ends can give us the largest square, an alternate approach could be to use two pointers starting at both the ends of the input array to iterate faster. At any step, whichever pointer gives us the bigger

square we add it to the result array and move to the next/previous number according to the pointer. For the above-mentioned Example-1, we will do something like this:

Squared sorted array 2

Javascript code below:-

```html
<script>
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;

}


document.write(`Squares: ${make_squares([-2, -1, 0, 2, 3])}<br/>`);

document.write(`Squares: ${make_squares([-3, -1, 0, 1, 2])}<br/>`);

</script>
```

Time complexity

The time complexity of the above algorithm will be **O(N)** as we sorting and squaring the array in the same iteration.

Space complexity

The space complexity of the above algorithm will also be **O(N);** this space will be used for the output array.

Comments

comments

Golibrary (https://www.golibrary.co/author/goliwadapaav4348/)

An avid reader, responsible for generating creative content ideas for golibrary.co. His interests include algorithms and programming languages. Blogging is a hobby and passion.

(https://www.gol

IgIuR6v3zyBBGNimw1DC bUoEv6PgiC7qkBhKEr45 MGaWNZ6%2FEJmVdGI ymcBtkHFJZhl3UjrNztZPs FtHqctuSNP7UHSAmm4 5RXtuTjBzU0sf%2BAfMB %2FgEwsh7PTmgRb1ke 0f3JOq%2BIWbkAuRPbB s5Gd%2Ffe%2BxG2EoM 2fH8DfB)

emsy%2DRs0u&XmOBZd 7PF74sKYWTZUWIR%2F jG6Gezpy5MArF9wdMSM 91pW8htXeSR%2Fm6Vy KGydJ7k5I1IRVoaZcJCw7 ru1ULua0Qvy5rCWfZXhs MNtNB%2FumPg4VywPs Of%2Fn8UXv6lKWbdUzt UpkhTJThW8EoE%2FXv kPvhuSveRbZNTr3wdTjy u%2F9xaLpXxQ)

HTTPS://WWW.GOLIBRARY.CO/?P=3103)   g+ GOOGLE +

(HTTPS://PLUS.GOOGLE.COM/SHARE?URL=HTTPS://WWW.GOLIBRARY.CO/SQUARING-

A-SORTED-ARRAY/)   📌 PINTEREST (HTTP://PINTEREST.COM/PIN/CREATE/BUTTON/?

URL=HTTPS://WWW.GOLIBRARY.CO/SQUARING-A-SORTED-ARRAY/ -

HTTPS://WWW.GOLIBRARY.CO/?P=3103)

Tags :   algorithms (https://www.golibrary.co/tag/algorithms/)

array (https://www.golibrary.co/tag/array/)

javascript (https://www.golibrary.co/tag/javascript/)

sorted (https://www.golibrary.co/tag/sorted/)   square (https://www.golibrary.co/tag/square/)

<< What caused the big bang ?
(https://www.golibrary.co/what-caused-
the-big-bang/)

Remove duplicates from sorted array in place >> (https://www.golibrary.co/remove-duplicates-
from-sorted-array-in-place/)

RELATED POSTS

💬 0 / 👁 282

💬 0 / 👁 286

⌃⌃

[Algorithm to find all factors of very large number](https://www.golibrary.co/algorithm-to-find-all-factors-of-very-large-number/)

[Algorithm to find prime factors of very big numbers](https://www.golibrary.co/algorithm-to-find-prime-factors-of-very-big-numbers/)

34GW%2FhRtPg3NqLa0eRMgPo3%2FGP1ZUrrkLxe6YU7UTqN9ZW576GVJfskY9dA6U38%2FiqZYX%2BBBVRTw%2F27bdah%2FqMd%2BCjUpYmkfbVAjF0nWMsvhiNE7gv7jvy5e6Zc5MD%2FXTIO3QI%2F0X0pDbVPubgMW8Fo%2BZeHnB40VrHDTAy9NEcSMlvwou8dVCRPFqt0vxVpP7V5zsQbP0GfOIvtIKvZkmi0IMjQgKNPMiZctvK7oAM9u9QDb2XIbXzcFRTVuo1LvEU4NbYFDGG2tym0snUpR6nA59Oad%2BW6bDxr8cyCCq8HsGbJaDGjFjZWomIS02uEtuWFAE)9oEtpXfxkXWN7KH%2FWxvmg)

Creepiest Abandoned Amusement Parks Around the World (https://api.content.ad/Lib/TrackOutboundClick.aspx?hid=-1&wid=158180&uid=2627302&did=190439&type=3&s=ip-172-18-2-48.ec2.internal&c=EG&d=&aid=8DB38BA3-CF48-42E6-8AFD-457A1EFC7626&ik=2022030102_f0edf4e3a71433a0cb8dedc4f332731d&it=2022-03-01+02%3A13%3A29.977&hai=43414&campaignId=134154&advertiserId=7535&guid=58cddf74-f0be-4523-ba3b-2e34ef2d6ad5&ck=a08ec94ed129eb12f28015f9c3a98865&adi=default&t=false&ct=onpage&pan=4kzuPIaSadmJF5Iv%2FdLVow%3D%3D&oc=LDu0XkKQVHr0V0kxSUChRg%3D%3D&redirectUrl=Fs5hfcm8y%2BK86uQxMUBEa34Gw%2FhRtPg3NqP1ZUrrkLxe6YU7UT9dA6U38%2FiqZYXdah%2FqMd%2BCjUvhiNE7gv7jvy5e6Zc50X0pDbVPubgMW8DTAy9NEcSMlvwou85zsQbP0GfOIvtIKvZvK7oAM9u9QD

15 Hilarious (and Painful) Model Rocket Launch Fails (https://api.content.ad/Lib/TrackOutboundClick.aspx?hid=-1&wid=158180&uid=2634096&did=190439&type=3&s=ip-172-18-2-48.ec2.internal&c=EG&d=&aid=8DB38BA3-CF48-42E6-8AFD-457A1EFC7626&ik=2022030102_f0edf4e3a71433a0cb8dedc4f332731d&it=2022-03-01+02%3A13%3A29.977&hai=43414&campaignId=134154&advertiserId=7535&guid=58cddf74-f0be-4523-ba3b-2e34ef2d6ad5&ck=d821a8a492a5d6c0d862925808342876&adi=default&t=false&ct=onpage&pan=4kzuPIaSadmJF5Iv%2FdLVow%3D%3D&oc=LDu0XkKQVHr0V0kxSUChRg%3D%3D&redirectUrl=Fs5hfcm8y%2BK86uQxMUBEa8eKMgP0s%2FGqN9ZW576GVJfskY%2BBBVRTw%2F27bpYmkfbVAjF0nWMsMD%2FXTIO3QI%2Fo%2BZeHnB40VrHdVCRPFqt0vxVpP7Vkmi0IMjQgKNP

IUMtiRv2vK78AM9u9QD
b2XINbYFDGG2tym0snU
6bDxr8cyCCq8HsGbEtu
WFAE)

vXvpP7vKmIohMjQgRNP
MiZctbXzcFRTVuo1LvEU
4pR6nA59Oad%2BWJaD
GjFjZWomIS02u9oEtpXfx
kXWN7KH%2FWxvmg)

FACEBOOK FANS

Golibrary
(https://www.facebook.com/GolibraryIndia-
174547689350736/)

TAGS

AI (https://www.golibrary.co/tag/ai/)

algorithm
(https://www.golibrary.co/tag/algorithm/)

algorithms
(https://www.golibrary.co/tag/algorithms/)

array
(https://www.golibrary.co/tag/array/)

C++
(https://www.golibrary.co/tag/cpp/)

cloud
(https://www.golibrary.co/tag/cloud/)

devops
(https://www.golibrary.co/tag/devops/)

google
(https://www.golibrary.co/tag/google/)

javascript
(https://www.golibrary.co/tag/javascript/)

ABOUT US

Golibrary is an upcoming education startup aiming to act as a one stop shop for students , teachers and education institutes worldwide to interact and access a wide array of info online.

GET IN TOUCH

🏠 Sydney, Australia

✉ info@golibrary.co
(mailto:info@golibrary.co)

COLLABORATIVE EDUCATION

Golibrary aims to be a collaborative education and learning platform for individuals of all age groups.

Linux (https://www.golibrary.co/tag/linux/)

market (https://www.golibrary.co/tag/market/)

ML (https://www.golibrary.co/tag/ml/)

python (https://www.golibrary.co/tag/python/)

stock (https://www.golibrary.co/tag/stock/)

subarray (https://www.golibrary.co/tag/subarray/)

sum (https://www.golibrary.co/tag/sum/)

Golibrary.co   Privacy (privacy-policy)   Terms (terms)