Ace your JavaScript Interview. Get my ebook. 100 solved Javascript, 20 solved React, & 2 frontend system design questions (1160+ copies sold). Get a Free preview.

Advertisements

# Bubble sort using two stacks

Posted on September 13, 2019 | by Prashant Yadav

Posted in Algorithms, Sorting, Stack | Tagged Easy

Learn how to implement an algorithm to sort an array using two stacks and bubble sort paradigm in javascript.

## Example

```
Input:
10 2 45 22 -1 0

Output:
-1 0 2 10 22 45
```

## Implementation to bubble sort an array with two stacks.

- First push all the elements of the array in one stack.
- Then loop all the elements of the array and alternatively swap the elements using both the stacks.
- To swap keep on pushing elements in one stack till the top of the other stack is smaller than the element being pushed from first stack.
- If the element being pushed is smaller than top of other stack then swap them (as in bubble sort).
- And in the end (tricky step) get the top element from the stack and place it in the array at its position.
- Keep doing this alternatively for both the stacks

Practically prepare for your JavaScript interview

JavaScript Revision
JavaScript-Concept Based Problems
Data Structures
Algorithms
Machine Coding
Web Fundamentals

Advertisements

Copy

```
let bubbleSortWithStack = (arr, n = arr.length) => {
  //First stack
  let stack1 = new Stack();

  //Copy all the elements of
  //array in the stack
  for(let elm of arr){
    stack1.push(elm);
  }

  //Second stack
  let stack2 = new Stack();

  for(let i = 0; i < n; i++){

    if(i % 2 == 0){
      //Loop till stack1 has elements
      while(!stack1.isEmpty()){
        //Get the top element from the stack1
        let t = stack1.pop();

        //If stack2 is empty then add the
        //Element to the stack2
        if(stack2.isEmpty()){
          stack2.push(t);
        }else{
          //Else compare the top elements of stack1 and stack2
          //And swap them based on the sorting order
          if(stack2.peek() > t){
            let temp = stack2.pop();
            stack2.push(t);
            stack2.push(temp);
          }else{
            stack2.push(t);
          }
        }
      }

      // Tricky step
      arr[n-1-i] = stack2.pop();
    }else{
      //Loop till stack2 has elements
      while(!stack2.isEmpty()){

        //Get the top element from stack2
        let t = stack2.pop();

        //If stack1 is empty then
        //Then add the element in the stack2
        if(stack1.isEmpty()){
          stack1.push(t);
        }else{
          //Else compare the stack1 and stack2 top elements
          //And swap them based on the sorting order
          if(stack1.peek() > t){
            let temp = stack1.pop();
            stack1.push(t);
            stack1.push(temp);
          }else{
            stack1.push(t);
          }
```

```
        }
      }

      //Tricky step
      arr[n-1-i] = stack1.pop();
    }
  }
}
```

Copy

```
Input:
let arr = [10, 2, 45, 22, -1, 0];
bubbleSortWithStack(arr);
console.log(arr);

Output:
[-1, 0, 2, 10, 22, 45]
```

Time complexity: O(n ^ 2).

Space complexity: O(n).

Prepare for your **JavaScript Interview** practically on each Interview rounds and grab that job.

BEGIN LEARNING

## Recommended Posts:

Program to print the Collatz sequence in javascript.

FizzBuzz program in javascript

Longest common subsequence | Print all LCS

Find the largest prime factor

Find number of trailing zeros in factorial

3Sum closest

Selection sort in javascript

Convert Roman numeral to an integer

Program to check balanced parentheses

3 sum problem algorithm

Prev

Next

Advertisements

Handcrafted with ♥somewhere in **Mumbai**

© 2023 LearnersBucket | Prashant Yadav