GeeksforGeeks
A computer science portal for geeks

GeeksQuiz

Login

Home Algorithms DS GATE Interview Corner Q&A C C++ Java Books Contribute Ask a Q About

Array Bit Magic C/C++ Articles GFacts Linked List MCQ Misc Output String Tree Graph

Pancake sorting

Given an an unsorted array, sort the given array. You are allowed to do only following operation on array.

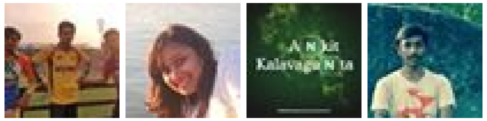
```
flip(arr, i): Reverse array from 0 to i
```

Unlike a traditional sorting algorithm, which attempts to sort with the fewest comparisons possible, the goal is to sort the sequence in as few reversals as possible.

The idea is to do something similar to Selection Sort. We one by one place maximum element at the end and reduce the size of current array by one.

Following are the detailed steps. Let given array be arr[] and size of array be n.
1) Start from current size equal to n and reduce current size by one while it’s greater than 1. Let the current size be curr_size. Do following for every curr_size
……a) Find index of the maximum element in arr[0..curr_szie-1]. Let the index be ‘mi’
……b) Call flip(arr, mi)
……c) Call flip(arr, curr_size-1)

See following video for visualization of the above algorithm.



Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms

```
/* A C++ program for Pancake Sorting */
#include <stdlib.h>
#include <stdio.h>

/* Reverses arr[0..i] */
void flip(int arr[], int i)
{
    int temp, start = 0;
    while (start < i)
    {
        temp = arr[start];
        arr[start] = arr[i];
        arr[i] = temp;
        start++;
        i--;
    }
}

/* Returns index of the maximum element in arr[0..n-1] */
int findMax(int arr[], int n)
{
   int mi, i;
   for (mi = 0, i = 0; i < n; ++i)
       if (arr[i] > arr[mi])
              mi = i;
   return mi;
}
```

Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

Popular Posts

All permutations of a given string
Memory Layout of C Programs
Understanding “extern” keyword in C
Median of two sorted arrays
Tree traversal without recursion and without stack!
Structure Member Alignment, Padding and Data Packing

```
// The main function that sorts given array using flip operations
int pancakeSort(int *arr, int n)
{
    // Start from the complete array and one by one reduce current siz
    for (int curr_size = n; curr_size > 1; --curr_size)
    {
        // Find index of the maximum element in arr[0..curr_size-1]
        int mi = findMax(arr, curr_size);

        // Move the maximum element to end of current array if it's no
        // already at the end
        if (mi != curr_size-1)
        {
            // To move at the end, first move maximum number to beginn
            flip(arr, mi);

            // Now move the maximum number to end by reversing current
            flip(arr, curr_size-1);
        }
    }
}

/* A utility function to print an array of size n */
void printArray(int arr[], int n)
{
    for (int i = 0; i < n; ++i)
        printf("%d ", arr[i]);
}

// Driver program to test above function
int main()
{
    int arr[] = {23, 10, 20, 11, 12, 6, 7};
    int n = sizeof(arr)/sizeof(arr[0]);

    pancakeSort(arr, n);

    puts("Sorted Array ");
    printArray(arr, n);

    return 0;
}
```

Output:

```
Sorted Array
```

Data Packing

Intersection point of two Linked Lists

Lowest Common Ancestor in a BST.

Check if a binary tree is BST or not

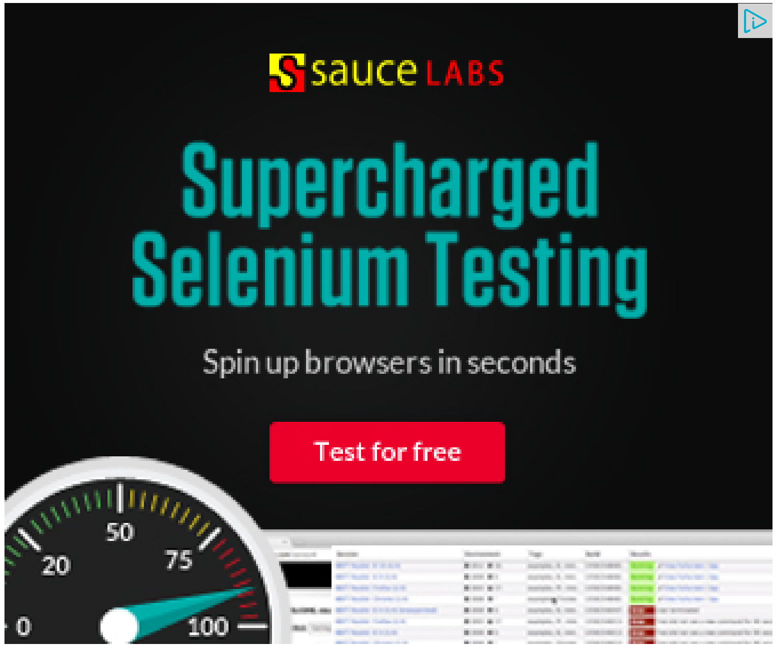
Sorted Linked List to Balanced BST

```
6 7 10 11 12 20 23
```

Total O(n) flip operations are performed in above code. The overall time complexity is O(n^2).

References:
http://en.wikipedia.org/wiki/Pancake_sorting

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



Related Tpoics:

- Remove minimum elements from either side such that 2*min becomes more than max
- Divide and Conquer | Set 6 (Search in a Row-wise and Column-wise Sorted 2D Array)
- Bucket Sort
- Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1
- Find the number of zeroes
- Find if there is a subarray with 0 sum
- Divide and Conquer | Set 5 (Strassen's Matrix Multiplication)
- Count all possible groups of size 2 or 3 that have sum as multiple of 3

705 Subscribe

Recent Comments

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 9 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 12 minutes ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 37 minutes ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 39 minutes ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

newCoder3006 Code without using while loop. We can do it...

Find subarray with given sum · 1 hour ago

34 Tweet 3 1

Writing code in comment? Please use **ideone.com** and share the link here.

22 Comments GeeksforGeeks

Sort by Newest ▾

Join the discussion…

Curious Wanderer • 7 months ago

I think Bill Gates came up with pancake sorting algorithm.

Reply • Share ›

rakitic • 10 months ago

@geeks...for each elements , we are considering two flips and one maximum

Reply • Share ›

Ronny ➦ rakitic • 9 months ago

@rakitic

But these are happening independent of each other.
O(n)+O(n)+O(n) = O(n)
O(n) for each of the element
So complexity is O(n^2) only

Reply • Share ›

rakitic • 10 months ago

for each elements , we consider two flips and one maximum ..should'nt be tot

Reply • Share ›

MITTAL • 10 months ago

My code is working as insertion sort and the complexity is O(n2)..

if not able to follow the code plz comment..

```
[sourcecode language="C++"]
/*
if(arr[0]>arr[1])
{
arr[0]^=arr[1];
arr[1]^=arr[0];
arr[0]^=arr[1];
}
for(int i=2;i<arr.size();i++)
{
if(arr[i]<arr[i-1])
{
flip(arr,i-1);
flip(arr,i);
flip(arr,i-1);
flip(arr,i-2);
i=i-2;
}
} */
```

Reply • Share ›

Jinyao Xu • a year ago

I don&#039t think your solution can find the optimized sorting steps(fewest flip

1 • Reply • Share ›

Prateek Sharma • a year ago

Python Code.......

```
def flip(a,ArrayIndex):
    reverseList =[]
    for i in range(ArrayIndex,-1,-1):
        reverseList.append(a[i])
    for i in range(len(reverseList)):
        a[i] = reverseList[i]

def pancakeSorting(a):
    arraylist =[]
    while len(a)>1:
        maxIndex = a.index(max(a))
        flip(a,maxIndex)
        flip(a,len(a)-1)
        arraylist.insert(0,a[-1])
        a.pop()
    print arraylist
def main():
    array= [45,7,3,89,123,56]
    pancakeSorting(array)
if __name__ == '__main__':
    main()
```

• Reply • Share ›

Theopaul • a year ago

I think 2 flips are enough
Array : 23, 10, 20, 11, 12, 6, 7
Consider the array sorted halfway.

Sorted array for 3 elements

10 , 20 , 23

Now we have to insert 11.

The binary search index will return position 1 (20)

flip from 20 to end ie (23)

The array becomes: 10 , 23 , 20 , 11 ....

Now flip from 23 to 11.

The resulting array would be: 10 , 11 , 20 , 23 similarly 12 comes we need to fl

Array becomes: 10 , 11 , 23 , 20 , 12

see more

^ | ⌄ • Reply • Share ›

gautam kumar ➦ Theopaul • a year ago

Read the definition of flip carefully.

```
/* Paste your code here (You may delete these lines if not wr
```

^ | ⌄ • Reply • Share ›

viki • a year ago

@Kumar: Will you please write the solution you just have proposed...:D

^ | ⌄ • Reply • Share ›

Kumar • a year ago

The above code runs in O(n^2) even after assuming that
flip operation runs in O(1), otherwise it will run in O(n^3).

Suppose flip operation runs in O(1) Can we do better ?
Can we make the above code to be run in O(nlogn) ??

Yes, We need to think of insertion sort using binary search rather than selectio

```
for(i=0;i<(a.size-1);i++) <---- O(n)
{
int key = a[i+1];
index = BinarySearch(a,key,0,i); <---- O(logn)
/* now we can apply flip operation here, think about it */
/* hint there will be three flip operation */
/* Let me know if you won&#039t get */
}
```

Since all flip operation will take O(1) by assumption
Total running time will be O(nlogn)

Reply • Share ›

kartik ➦ Kumar • a year ago

I have written solution using 4 flips.

```
 /* A C program for Pancake Sorting */
#include <stdlib.h>
#include <stdio.h>



/* A Binary Search based function to get index of ceiling of x
int ceilSearch(int arr[], int low, int high, int x)
{
  int mid;

  /* If x is smaller than or equal to the first element,
    then return the first element */
  if(x <= arr[low])
    return low;

  /* If x is greater than the last element, then return -1 */
```

see more

• Reply • Share ›

Kumar kartik • a year ago

Thanks KArthik,

I haven't tested but I'm sure it will work, thanks for coding it.

• Reply • Share ›

itreallyismE kartik • a year ago

Just a quick doubt. Doesn't binary search work for sorted array case?

• Reply • Share ›

itreallyismE itreallyismE • a year ago

Sorry about that. Got it.

• Reply • Share ›

viki Kumar • a year ago

@Kumar: How r u thinking dude ? Pancake sort runs in O(n^2) time ev
in O(n) time in worst case. correct yourself......

• Reply • Share ›

rakitic viki • 10 months ago

two flips each time = n^2 , max each time = n , no of elements

• Reply • Share ›

Kumar viki • a year ago

I said, if we assume that flip operation runs in O(1) then sorting
flip operation takes O(n) time) then sort will run take O(n^3).... p

Regarding implementation, I'll definitely send, let me code it firs

```
/* Paste your code here (You may delete these lines if
```

Reply • Share ›

gokul ➜ Kumar • a year ago

viki is correct on pointing it out... its O(n^2) even if flip ta
Please you understand the code carefully

```
/* Paste your code here (You may delete these li
```

Reply • Share ›

kartik ➜ gokul • a year ago

The problem mentioned by Kumar is different from the p
mentioned, imagine a hypothetical machine where flip ta
the flip operation takes O(1), you can actually sort the a

Reply • Share ›

kartik ➜ Kumar • a year ago

Goof question @Kumar, we will be publishing it as a separate post.

Reply • Share ›

saurabh ➜ kartik • a year ago

Goof ;)

1 • Reply • Share ›

Subscribe   Add Disqus to your site