Basic Problem Solving: Arrays and Objects

Relevel
by Unacademy

List of Concepts Involved (10mins)

- Minimum and Maximum Element
- Maximum Profit Stock
- Reverse Array
- Sort Array having 0,1
- Reverse Subarray to Sort array

Minimum and Maximum Element

Problem – Given an array of numbers. Your task is to find the minimum and maximum element. For example –

Input – [2,3,15,6]

Output –
Minimum element - 1
Maximum element - 15

Approach – We can use linear search technique in this.

Steps -

1) Initialize min and max to minimum and maximum of first 2 elements
2) Iterate through array from index 3rd and update min and max respectively after comparison
3) Return the output

Implementation

Code-> https://jsfiddle.net/fnog50k3/

Maximum Profit Stock

Problem – Given an array denoting the cost of stock on different days. Your task is to find the maximum profit which can be earned after buying or selling the stock on that particular day. For example –

Input – `{100, 180, 260, 310, 40, 535, 695}`

Output – 865 (Buy stock on Day 0 and sell it on Day 3 and again buy on Day 4 and sell on Day 6)

Naive Approach – We can iterate through the array and buy and sell stock everyday to check if we are getting maximum profit or not. If we get more profit, we can update our maximum profit so far.

We can use nested loop in this case to get the desired value

Maximum Profit Stock

Code Link - https://jsfiddle.net/agr7nudz/

```
function maxProfit( price, start, end)
{
  if (end <= start)
    return 0;

  let profit = 0;

  for (let i = start; i < end; i++) {
    for (let j = i + 1; j <= end; j++) {


      if (price[j] > price[i]) {

        let curr_profit = price[j] - price[i]
                + maxProfit(price, start, i - 1)
                + maxProfit(price, j + 1, end);

        profit = Math.max(profit, curr_profit);
      }
    }
  }
  return profit;
}
```

Maximum Profit Stock

Efficient Approach –Instead of using nested loops, we will use a single loop here.

STEPS - {100, 180, 260, 310, 40, 535, 695}

1) Initialize maximumProfit = 0;
2) Iterate from 1 to length of array
3) Check if current stock price is greater than previous price
4) If yes, save difference of current and previous to the maximum Profit

Maximum Profit Stock

Code Link - https://jsfiddle.net/60L1gmfa/

```
function maxProfit(prices , size) {

  var maxProfit = 0;

  for (i = 1; i < size; i++)
    if (prices[i] > prices[i - 1])
      maxProfit += prices[i] - prices[i - 1];
  return maxProfit;
}

  var price = [ 200, 280, 360, 410, 140, 535, 695 ];
  var n = price.length;

  console.log(maxProfit(price, n));
```

Reverse Array

Problem – Given an array of numbers. You need to reverse the array. For example –
Input – [1,2,3,4,5]
Output – [5,4,3,2,1]

Approach 1 – We can use swapping operation to get the desired output. Let's see each step –

1) Initilaize two pointers - start=0, end = n-1 where n = number of elements in array
2) Swap elements present at start and end
3) Start = start+1, end = end -1
4) Repeat step1 till start <= end

Reverse Array

Time Complexity –

If there are N elements in array. Then complexity will be O(N)

Code Link -> https://jsfiddle.net/4u9fLg2b/

```
function rvereseArray( arr, start,  end)
{
  var temp;

  while (start < end)
  {
    temp = arr[start];
    arr[start] = arr[end];
    arr[end] = temp;
    start = start + 1;
    end = end - 1;
  }
}

function printArray( arr,size)
{
  for ( i = 0; i < size; i++)
    console.log(arr[i] + " ");

}
  arr  = [1, 2, 3, 4, 5, 6];
  printArray(arr, 6);
  rvereseArray(arr, 0, 5);
  console.log("Reversed array is - ");
  printArray(arr, 6);
```

Reverse Array

Approach 2 – We can use recursion to get the desired output. Let's see each step –

1) Initilaize two pointers - start=0, end = n-1 where n = number of elements in array
2) Swap elements present at start and end
3) Recursively call reverseArray function

Reverse Array

Time Complexity –
If there are N elements in array. Then complexity will be O(N)

Code Link ->
https://jsfiddle.net/9whmqbgf/

```
function reverseArray( arr , start ,  end)
{
      var temp;
      if (start >= end)
          return;
      temp = arr[start];
      arr[start] = arr[end];
      arr[end] = temp;
      reverseArray(arr, start+1, end-1);
}

function printArray( arr ,size)
{
  for ( i = 0; i < size; i++)
    console.log(arr[i] + " ");

}
  arr  = [1, 2, 3, 4, 5, 6];
  printArray(arr, 6);
  reverseArray(arr, 0, 5);
  console.log("Reversed array is - ");
  printArray(arr, 6);
```

Sort Array of 0,1

Given an array having 0,1 as input.We need to write a function that sorts the array in ascending order without using inbuilt sort function
Expected time complexity :O(n)
Expected space complexity:O(1)

Example:
Input: {0, 1, 1,0, 0, 1}
Output: {0, 0, 0, 1, 1, 1}

Input:{0,0,1,1,0,1,0}
Output:{0,0,0,0,1,1,1}

Approach:

We divide the array into three parts .Part one containing the zeros,second part containing the values which can be zero or one and the last part containing one.If the element in second part is zero will reduce the size of second part,if the element is one will move it to the third part and reduce the size of second part.

Algorithm

1)We will have two indices mid=0,end=N and there are three parts

(0,mid):the elements here will be 0

(mid,end):the elements here can be 0,1

(end,N):the elements here will be 1

Where N = size of the array

2)We will be traversing the array from start to end until mid is less than end

3)If the mid element is 0,increment mid by one

4)If the mid element is 1,will swap with the element at index end and decrement the value of end by one

6)We will continue doing this till the mid value is less than the end.

Implementation:

Code:https://jsfiddle.net/041zoLwu/1/

```
var input = [1,1,1,0,0,1,1,0,0,0,1,0,1,0,1];
var size = input.length;
// function to sort an array containing 0,1, and return the sorted arr
ay
function sortZeroOne(input, size) {
  let mid = 0;
  let end = size-1;
  let swap = 0; //variable for swapping
  while (mid <= end) {
    if (input[mid] == 0) {
      mid++;
    } else {
      swap = input[end];
      input[end] = input[mid];
      input[mid] = swap;
      end--;
    }
  }
  return input;
}

var output = sortZeroOne(input, size);
console.log(output);
```

Complexity Analysis:

Time Complexity :O(n)
Only one iteration of the array is made
Space Complexity:O(1)
No extra space is required

Reverse Subarray to Sort Array

Problem – Given an array of numbers. You need to find if reversing a subarray can sort the array. For example –
Input – [1,2,5,4,3]
Output – true (reversing {5,4,3} will sort the array)

Approach –
Intuition - If array is having structure like first increasing, then decreasing and then again increasing, then we can say that after reversing the decreasing subarray, we can get our sorted array.

Let's see each step –

1) Find Increasing subarray at the start
2) Find decreasing subarray
3) At last, check for increasing subarray
4) In third step, if we are unable to find elements, this also gives us case where we can just reverse decreasing subarray of step 2 amd get sorted array

Reverse Subarray to Sort Array

Time Complexity –

If there are N elements in array. Then complexity will be O(N)

Code Link -> https://jsfiddle.net/kbne4mx7/

```
function checkSorted(arr, n) {
  if (n == 1)
    return true;
  let i;
  for (i = 1; i < n && arr[i - 1] < arr[i]; i++);
  if (i == n)
    return true;

  let j = i;
  while (j < n && arr[j] < arr[j - 1]) {
    if (i > 1 && arr[j] < arr[i - 2])
      return false;
    j++;
  }
  if (j == n) // when step 3 elements not present
    return true;

  let k = j;
  if (arr[k] < arr[i - 1])
    return false;
  while (k > 1 && k < n) {
    if (arr[k] < arr[k - 1])
      return false;
    k++;
  }
  return true;
}
```

MCQ

1)What is the output of relevle[1] of the following code?
 var relevel = [3,,6];

a)Undefined
b)Error
c)Null
d)0

Ans:a
Array has null value when no value is entered but if value at some index is omitted it takes undefined as value

2)What is the functionality of the pop method in an array?

a)decreases the length of array by one
b)increases the length of array by one
c)returns the deleted element
d)deletes the first element of array

Ans:a
Pop function delete the last element and hence decrease the array length by one

3)What is the output of the following code?

```
var one = [1,2,3];
var two = [4,5,6];
var ans = one.concat(two);
console.log(ans);
```

a)4,5,6,1,2,3
b)1,2,3,4,5,6
c)Error
d)Null

Ans:b
Concat is a predefined function in js which is used to combine elements of two arrays

4)What is the output of javascript code?

```
var sum = 0;
var input = [10,20,30,80];

input.forEach((elem) => {
                    sum+=elem;
          }
);
```

a)140
b)100
c)80
d)60

Ans:a
forEach work in the same way as for loop and iterates through each element of the array

5)What is the output of the following code?

```
var a = [1,2,4,5,6];
console.log(a.slice(0,2));
```

a)[1,2]
b)[5,6]
c)[2,4]
d)[1,2,4];

Ans:a
slice is a predefined function in javascript which is used to return elements of an array from starting point to the ending point as mentioned in function argument.

Practice Questions

1. Write code for creating a new sorted array from two sorted arrays in O(n+m) time complexity where n,m are the size of the unsorted arrays.
2. Write code for finding the smallest element in an array.

Upcoming Class Teaser

- Pick problems from codeforces also for basic arrays
- Anagram
- Print frequency of elements in string
- First non repeating character
- Subarray with sum 0
- Subarray with sum x
- Longest consecutive sequence
- More problems around array and objects (for objects hashing problems can be used)

THANK YOU