<https://javarevisited.blogspot.com/2015/06/top-20-array-interview-questions-and-answers.html>

|  |  |
| --- | --- |
| The third Largest element |  |
| Removing Duplicate in Unsorted array  O(n2) |  |
| Reverse Array O(n/2) |  |
| Sqare root of a number |  |
| Find duplicate in an array  **Time Complexity:** O(n) **Auxiliary Space:** O(1) | **public** **class** FindDuplicate {  **public** **static** **void** main(String[] args) {  FindDuplicate duplicate = **new** FindDuplicate();  **int** arr[] = { 1, 2, 8, 3, 1, 3, 6, 6, 8 };  **int** arr\_size = arr.length;  duplicate.printRepeating(arr, arr\_size);  }  // Function to print duplicates  **void** printRepeating(**int** arr[], **int** size) {  **int** i;  System.***out***.println("The repeating elements are : ");  **for** (i = 0; i < size; i++) {  **if** (arr[Math.*abs*(arr[i])] >= 0)  arr[Math.*abs*(arr[i])] = -arr[Math.*abs*(arr[i])];  **else**  System.***out***.print(Math.*abs*(arr[i]) + " ");  }  }  } |
| HCF of 2 nos |  |
| HCF |  |
| LCM | N1\* N2/HCF |
| How to find the missing number in integer array of 1 to 100? | 1) Calculate the sum of all numbers stored in the array  2) Subtract the sum Formula : n \* (n + 1) / 2. |
| **How to check if array contains a number in Java?** | This problem is essentially how to search an element in the array. There are two options sequential search or binary search. You should ask the interviewer about whether an array is sorted or not, if the array is sorted then you can use binary search to check if given number is present in an array or not. The complexity of binary search is O(logN). BTW, if interviewer says that array is not sorted then you can still sort and perform binary search otherwise you can use sequential search. Time complexity of sequential search in array is O(n). |
| How to find largest and smallest number in unsorted array? |  |
| How to find all pairs on integer array whose sum is equal to given number?  complexity of this solution is O(n^2) |  |
| O(n) |  |
|  | A more efficient in-place solution would be to sort the array and use two pointers to scan through array from both direction i.e. beginning and end. If sum of both the values are equal to given number then we output the pair and advance them. If the sum of two numbers is less than k then we increase the left pointer, else if the sum is greater than k we decrement the right pointer, until both pointers meet at some part of the array. The complexity of this solution would be O(NlogN) due to sorting. Remember to use a in-place sorting algorithm like [quicksort](http://java67.blogspot.sg/2014/07/quicksort-algorithm-in-java-in-place-example.html) to sort the array as we don't have additional space. Thankfully, Arrays.sort() method uses a two pivot quicksort algorithm to sort array of primitives. |
| Arrays.sort() method uses a two pivot quicksort algorithm to sort array of primitives. |  |
| 7. Write a program to remove duplicates from array in Java? |  |
| Write a program to find intersection of two sorted arrays in Java? |  |
| There is an array with every element repeated twice except one. Find that element? | A better solution is to use hashing. 1) Traverse all elements and put them in a hash table. Element is used as key and count of occurrences is used as value in hash table. 2) Traverse the array again and print the element with count 1 in hash table. This solution works in O(n) time, but requires extra space. |
| XOR operation |  |
|  |  |
| 11. How to find kth smallest element in unsorted array?  12. How to find kth largest element in unsorted array? | A simple solution is to sort the given array using a O(N log N) sorting algorithm like [Merge Sort](http://geeksquiz.com/merge-sort/), [Heap Sort](http://geeksquiz.com/heap-sort/), etc and return the element at index k-1 in the sorted array.  Time Complexity of this solution is O(N Log N) |
| How to find common elements in three sorted array? |  |
|  | Use map,  Store all elements by a sing le for loop , and increase value if key is duplicate.  So finally return the the keys with valus 3. |
| Merge 2 sorted array |  |
| How to reverse array in place in Java? |  |
|  | **29. Difference between array and linked list data structure?** ([answer](http://javarevisited.blogspot.sg/2013/07/difference-between-array-and-linked-list-java.html)) This is a theoretical questions from phone interviews. There are several differences between array and linked list e.g. array stores element in contiguous memory location while linked list stores at random places, this means linked list better utilizes the places. Consequently, its possible to have large linked list in limited memory environment compare to array of same size. Advantage of using array is random access it provides if you know the index, while in linked list you need to search an element by traversing which is O(n) operation. |
|  |  |
|  |  |