

Sort And Searching Job Interview Questions And Answers



Interview Questions Answers

<https://interviewquestionsanswers.org/>

About Interview Questions Answers

Interview Questions Answers . ORG is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on Sort And Searching will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit [Sort And Searching Interview Questions And Answers](#) to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in Sort And Searching category. To ensure quality, each submission is checked by our team, before it becomes live. This [Sort And Searching Interview preparation PDF](#) was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
<https://twitter.com/InterviewQA>

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

Interview Questions Answers.ORG Team
<https://InterviewQuestionsAnswers.ORG/Support@InterviewQuestionsAnswers.ORG>



Sort And Searching Interview Questions And Answers Guide.

Question - 1:

What is Bitonic search?

Ans:

An array is bitonic if it is comprised of an increasing sequence of integers followed immediately by a decreasing sequence of integers. Given a bitonic array a of N distinct integers, describe how to determine whether a given integer is in the array in $O(\log N)$ steps. Hint: find the maximum, then binary search in each piece.

[View All Answers](#)

Question - 2:

What is Range search?

Ans:

Given a database of all tolls collected in NJ road system, devise a scheme to answer queries of the form: extract sum of all tolls collected in a given time interval. Use a Toll data type that implements the Comparable interface, where the key is the time that the toll was collected.

[View All Answers](#)

Question - 3:

Do you know why doesn't the Java library use a randomized version of quicksort?

Ans:

At the very least, the library should cutoff to some guaranteed $N \log N$ algorithm if it "realizes" it is in trouble. Perhaps to avoid side effects. Programmers may want their libraries to be deterministic for debugging. But the library only uses quicksort for primitive types when stability is not an issue, so the programmer probably wouldn't notice the randomness, except in running time.

[View All Answers](#)

Question - 4:

Tell me are there implementations for sorting and searching in the Java library?

Ans:

Yes. The Java library `java.util.Arrays` contains the methods `Arrays.sort()` and `Arrays.binarySearch()` that implement mergesort and binary search for Comparable types and a sorting implementation for primitive types based on a version of the quicksort algorithm, which is faster than mergesort and also sorts an array in place (without using any extra space). `SystemSort.java` illustrates how to use `Arrays.sort()`.

[View All Answers](#)

Question - 5:

What is Quicksort?

Ans:

Quick sort is a divide-and-conquer method for sorting. It works by partitioning an array of elements into two parts, then sorting the parts independently. As we shall see, the precise position of the partition depends on the initial order of the elements in the input file.

The crux of the method is the partitioning process, which rearranges the array to make the following three conditions hold:

- * The element $a[i]$ is in its final place in the array for i .
- * None of the elements $a[\text{left}], \dots, a[i-1]$ is greater than $a[i]$.
- * None of the elements in $a[i+1], \dots, a[\text{right}]$ is less than $a[i]$.

[View All Answers](#)

Question - 6:

What is Reduction to sorting method?

Ans:

A problem A reduces to a problem B if we can use a solution to B to solve A . Designing a new divide-and-conquer algorithm from scratch is sometimes akin to solving a puzzle that requires some experience and ingenuity, so you may not feel confident that you can do so at first. But it is often the case that a simpler approach is effective: given a new problem that lends itself to a quadratic brute-force solution, ask yourself how you would solve it if the data were sorted in some way. For



example, consider the problem of determining whether the elements in an array are all different. This problem reduces to sorting because we can sort the array, then make a linear pass through the sorted array to check whether any entry is equal to the next (if not, the elements are all different.)

[View All Answers](#)

Question - 7:

What is Mergesort?

Ans:

To develop a faster sorting method, we use a divide-and-conquer approach to algorithm design that every programmer needs to understand. This nomenclature refers to the idea that one way to solve a problem is to divide it into independent parts, conquer them independently, and then use the solutions for the parts to develop a solution for the full problem. To sort an array with this strategy, we divide it into two halves, sort the two halves independently, and then merge the results to sort the full array. This method is known as mergesort.

[View All Answers](#)

Question - 8:

What is Insertion sort?

Ans:

Insertion sort is a brute-force sorting algorithm that is based on a simple method that people often use to arrange hands of playing cards. Consider the cards one at a time and insert each into its proper place among those already considered (keeping them sorted).

[View All Answers](#)

Question - 9:

Explain exception filter?

Ans:

We now use binary search to solve the existence problem: is a given key in a sorted database of keys? For example, when checking the spelling of a word, you need only know whether your word is in the dictionary and are not interested in the definition. In a computer search, we keep the information in an array, sorted in order of the key. The binary search code in BinarySearch.java differs from our other applications in two details. First, the file size N need not be a power of two. Second, it has to allow the possibility that the item sought is not in the array. The client program implements an exception filter: it reads a sorted list of strings from a file (which we refer to as the whitelist) and an arbitrary sequence of strings from standard input and prints those in the sequence that are not in the whitelist.

[View All Answers](#)

Question - 10:

How to search binary in a sorted array?

Ans:

During much of the last century people would use a publication known as a phone book to look up a person's phone number. Entries appear in order, sorted by a key that identifies it (the person's name) in both cases. A brute-force solution would be to start at the beginning, examine each entry one at a time, and continue until you find the name. No one uses that method: instead, you open the book to some interior page and look for the name on that page. If it is there, you are done; otherwise, you eliminate either the part of the book before the current page or the part of the book after the current page from consideration and repeat.

[View All Answers](#)

Question - 11:

How to Inverting a function in Sort And Searching?

Ans:

As an example of the utility of binary search in scientific computing, we revisit a problem that we consider the problem of inverting an increasing function. To fix ideas, we refer to the Gaussian distribution \hat{I} when describing the method. Given a value y , our task is to find a value x such that $\hat{I}(x) = y$. In this situation, we use real numbers as the endpoints of our interval, not integers, but we use the same essential method as for guessing a hidden integer: we halve the size of the interval at each step, keeping x in the interval, until the interval is sufficiently small that we know the value of x to within a desired precision \hat{I} . We start with an interval (lo, hi) known to contain x and use the following recursive strategy:

* Compute $m = lo + (hi - lo) / 2$

* Base case: If $(hi - lo)$ is less than \hat{I} , then return m as an estimate of x

* Recursive step: otherwise, test whether $\hat{I}(m) < y$. If so look for x in (lo, m) ; if not look for x in (m, hi)

The key to this method is the idea that the function is increasing - for any values a and b , knowing that $\hat{I}(a) < \hat{I}(b)$ tells us that $a < b$, and vice versa. In this context, binary search is often called bisection search because we bisect the interval at each stage.

[View All Answers](#)

Question - 12:

Explain Binary representation?

Ans:

Binary search is nearly the same computation as converting a number to binary! Each guess determines one bit of the answer. In our example, the information that the number is between 0 and 127 says that the number of bits in its binary representation is 7, the answer to the first question tells us the value of the leading bit, the answer to the second question tells us the value of the next bit, and so forth. For example, if the number is 77, the sequence of answers no yes yes no no yes no immediately yields 1001101, the binary representation of 77.

[View All Answers](#)

Question - 13:

What is Linear-logarithm chasm?

Ans:



The alternative to using binary search is to guess 0, then 1, then 2, then 3, and so forth, until hitting the hidden number. We refer to such an algorithm as a brute-force algorithm: it seems to get the job done, but without much regard to the cost (which might prevent it from actually getting the job done for large problems). In this case, the running time of the brute-force algorithm is sensitive to the input value, but could be as much as N and has expected value $N/2$ if the input value is chosen at random. Meanwhile, binary search is guaranteed to use no more than $\lg N$ steps.

[View All Answers](#)

Question - 14:

Explain quick sort?

Ans:

Quick sort is one the fastest sorting algorithm used for sorting a list. A pivot point is chosen. Remaining elements are partitioned or divided such that elements less than the pivot point are in left and those greater than the pivot are on the right. Now, the elements on the left and right can be recursively sorted by repeating the algorithm.

[View All Answers](#)

Question - 15:

What is bubble sort algorithm in data structure sort and searching?

Ans:

Bubble sort algorithm is used for sorting a list. It makes use of a temporary variable for swapping. It compares two numbers at a time and swaps them if they are in wrong order. This process is repeated until no swapping is needed. The algorithm is very inefficient if the list is long.

E.g. List: - 7 4 5 3

1. 7 and 4 are compared
2. Since $4 < 7$, 4 is stored in a temporary variable.
3. the content of 7 is now stored in the variable which was holding 4
4. Now, the content of temporary variable and the variable previously holding 7 are swapped.

[View All Answers](#)

Question - 16:

Explain what is binary search?

Ans:

Binary search is most useful when the list is sorted. In binary search, element present in the middle of the list is determined. If the key (number to search) is smaller than the middle element, the binary search is done on the first half. If the key (number to search) is greater than the middle element, the binary search is done on the second half (right). The first and the last half are again divided into two by determining the middle element.

[View All Answers](#)

Question - 17:

Explain what is linear search?

Ans:

Linear search is the simplest form of search. It searches for the element sequentially starting from the first element. This search has a disadvantage if the element is located at the end. Advantage lies in the simplicity of the search. Also it is most useful when the elements are arranged in a random order.

[View All Answers](#)

Question - 18:

Tell me why might quick sort might be better than merge sort

Ans:

Quick sort also preserves order. Merge sort has undefined behavior wrt order.

[View All Answers](#)

Question - 19:

Tell me which of the following maintain index associations?

â€¢ ksort

â€¢ asort

â€¢ sort

Ans:

These are php sorting methods.

ksort and asort maintain index associations and sort from lowest to highest. ksort sorts keys.

sort doesn't maintain index associations.

[View All Answers](#)

Question - 20:

Explain which of the following is true about asort?

â€¢ Sorts highest to lowest by value maintaining key association.

â€¢ Sorts lowest to highest by key maintaining key association.

â€¢ Sorts highest to lowest by key, re-indexing the array.

â€¢ Sorts lowest to highest by value, re-indexing the array.

Ans:

Sorts lowest to highest by key maintaining key association.



asort()

This function sorts an array such that array indices maintain their correlation with the array elements they are associated with. This is used mainly when sorting associative arrays where the actual element order is significant.

[View All Answers](#)

Question - 21:

How to sort 1 million floating point numbers?

Ans:

Radix Sort.. easily can be done.. user bitwise operations to bucket the numbers.. same algorithm can be used for negative and positive mix of fp numbers with some minor modification to the initial list

[View All Answers](#)

Question - 22:

What is Binary Search Tree and explain its time complexity?

Ans:

Traverse: $O(n)$. Coz it would be visiting all the nodes once.

Search : $O(\log n)$

Insert : $O(\log n)$

Delete : $O(\log n)$

Binary Search is a searching algorithm that is used on a certain data structure (ordered array) to find a if an element is within the array through a divide a conquer technique that takes the middle value of the array and compares it to the value in question. If the value is less, then compare the value in question to the lower mid-value and if greater, vice versa until you find the value or determine that it is none of the elements within the array. All a Binary Search Tree is a visual concept that allows one to see this divide a conquer technique in an easier way. For example, given : Array A = {1,2,4,6,10,20,35} we ask if the number 35 is there. First you would compare 20 to 6. $20 <, >, = 6$? Greater than, so compare the mid value of upper range. $35 >, <, = 20$? greater than, compare next greater mid value. $35 >, <, = 35$? Equal. So it has been search with only three comparisons, with the worst time complexity of $O(\log(n))$.

[View All Answers](#)

Question - 23:

What is Mergesort and Hashtable?

Ans:

In short:

Mergesort is a sorting algorithm that follows the paradigm of: divide and conquer:

1) recursively split the array in 2

2) until the array length is 1 (or the pointers start and end are equal)

3) merge the sorted array an return the array sorted

[View All Answers](#)

Question - 24:

Do you know what is linear search?

Ans:

Linear search is the simplest form of search. It searches for the element sequentially starting from the first element. This search has a disadvantage if the element is located at the end. Advantage lies in the simplicity of the search. Also it is most useful when the elements are arranged in a random order.

[View All Answers](#)

Question - 25:

Explain binary search?

Ans:

Binary search is most useful when the list is sorted. In binary search, element present in the middle of the list is determined. If the key (number to search) is smaller than the middle element, the binary search is done on the first half. If the key (number to search) is greater than the middle element, the binary search is done on the second half (right). The first and the last half are again divided into two by determining the middle element.

[View All Answers](#)

Question - 26:

What is bubble sort algorithm?

Ans:

Bubble sort algorithm is used for sorting a list. It makes use of a temporary variable for swapping. It compares two numbers at a time and swaps them if they are in wrong order. This process is repeated until no swapping is needed. The algorithm is very inefficient if the list is long.

E.g. List: - 7 4 5 3

1. 7 and 4 are compared

2. Since $4 < 7$, 4 is stored in a temporary variable.

3. the content of 7 is now stored in the variable which was holding 4

4. Now, the content of temporary variable and the variable previously holding 7 are swapped.

[View All Answers](#)

Question - 27:



Tell me what is quick sort?

Ans:

Quick sort is one the fastest sorting algorithm used for sorting a list. A pivot point is chosen. Remaining elements are portioned or divided such that elements less than the pivot point are in left and those greater than the pivot are on the right. Now, the elements on the left and right can be recursively sorted by repeating the algorithm.

[View All Answers](#)

Data Structure Most Popular & Related Interview Guides

- 1 : [Data Center Manager Interview Questions and Answers.](#)
- 2 : [Creative UI/UX Designers Interview Questions and Answers.](#)
- 3 : [Data Structure Linked list Interview Questions and Answers.](#)
- 4 : [Data Structure Arrays Interview Questions and Answers.](#)
- 5 : [Data Architect Interview Questions and Answers.](#)

Follow us on FaceBook

www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter

<https://twitter.com/InterviewQA>

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team

[https://InterviewQuestionsAnswers.ORG/
support@InterviewQuestionsAnswers.ORG](https://InterviewQuestionsAnswers.ORG/support@InterviewQuestionsAnswers.ORG)