

OS Data Structures 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 OS Data Structures 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 [OS Data Structures 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 OS Data Structures category. To ensure quality, each submission is checked by our team, before it becomes live. This [OS Data Structures 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>



OS Data Structures Interview Questions And Answers Guide.

Question - 1:

What is a complexity of linear search, binary search?

Ans:

In linear search each element in the array should be checked until the required element got searched whereas in binary search array is divided into two and required element is searched

[View All Answers](#)

Question - 2:

What is AVL tree?

Ans:

An AVL tree is a self-balancing binary search tree, and it was the first such data structure to be invented. In an AVL tree, the heights of the two child subtrees of any node differ by at most one. Lookup, insertion, and deletion all take $O(\log n)$ time in both the average and worst cases, where n is the number of nodes in the tree prior to the operation. Insertions and deletions may require the tree to be rebalanced by one or more tree rotations.

[View All Answers](#)

Question - 3:

Explain real world example of polymorphism and encapsulation?

Ans:

Real world example for polymorphism can be start mechanism of bike, in which you will have same method start() but it may be either by kick start or button start.

Example for encapsulation can be a stack or queue doesn't matter how it is implemented internally with linked list or array

[View All Answers](#)

Question - 4:

What is difference between the run time polymorphism and compile time polymorphism and about virtual function?

Ans:

Compile time polymorphism (Static polymorphism) means basically those language structure which will cause the compiler to produce code at the compile-time. That is, the compiler is well aware that what code is to be generated at the compile-time itself. Ex: overloading of operators, functions.

Run time Polymorphism (Dynamic Polymorphism) means that the compiler is unaware what code is to be generated so it binds the possible code and let the program decide it at the run-time itself. Ex: the virtualness of a class member or the entire class itself.

[View All Answers](#)

Question - 5:

Why enum can not be used directly with printf function?



Ans:

enum is not an basic data type like int, float and all it is a user defined data type, and printf function works only with basic data type, we've overload printf function to make it work for user defined data types :)

[View All Answers](#)

Question - 6:

What is R-B tree?

Ans:

A red black tree is a binary tree where

1. every node has color.
2. root node is always black
3. the child of a black node is either black or red
4. both the child nodes of every red node must be black
5. all the leaves must be black

[View All Answers](#)

Question - 7:

What is the different between B-tree and B+ tree?

Ans:

It's all about branching factor. Because of the way B+-Trees store records (called "satellite information") at the leaf level of the tree, they maximize the branching factor of the internal nodes. High branching factor allows for a tree of lower height. Lower tree height allows for less disk I/O. Less disk I/O theoretically means better performance

In a B- tree you can store both keys and data in the internal/leaf nodes. But in a B+ tree you have to store the data in the leaf nodes only.

A B+ - Tree is in the form of a balanced tree in which every path from the root of the tree to a leaf of the tree is the same length.

Each nonleaf node in the tree has between $\lceil n/2 \rceil$ and n children, where n is fixed.

B+ - Trees are good for searches, but cause some overhead issues in wasted space.

[View All Answers](#)

Question - 8:

Tell me how to search an element in sorted linked list with time complexity is $O(\log n)$?

Ans:

we can use the binary search algorithm for this problem because this searching algorithm has $O(\log n)$ performance in both worse and average case.

[View All Answers](#)

Question - 9:

Do you know how to find the number of possible tree in the given tree?

Ans:

number of possible tree = $(2^{\text{power } n}) - n$.

for example:

A tree contain three node.

so $n=3$.

possible tree = $8 - 3 = 5$.

[View All Answers](#)

Question - 10:

Tell me why do tree always takes $O(\log n)$ time?

Ans:

Tree always takes $O(\log n)$ time because tree has height is $(\log n)$.

[View All Answers](#)

Question - 11:

Explain applications of stacks and their uses?

Ans:

Keeping track of nested invocation calls in a procedural programming language, such as C/C++.

Each function call results in a new entry being placed into the program run-time stack. This new

entry contains memory space for local variables (which can

grow dynamically) and for a return

pointer to the instruction in the function that invoked the current function (caller/callee). As

functions terminate, their stack entry is "popped out," with the return values written to the proper location in the caller.

Since nested procedural/ function invocation levels are



entered and exited in LIFO order, a stack is the most appropriate data structure to handle this functionality.

Evaluating arithmetic expressions.

Stacks can be used to parse arithmetic expressions and evaluate them efficiently, as we shall see as part of this assignment.

To eliminate the need for direct implementation of recursion.

As recursive function calls require a lot of overhead, it is often the case that recursive algorithms are "unrolled" into non-recursive ones. Since recursive calls are entered/exited in LIFO order the use of stacks to mimic recursion is a natural choice.

[View All Answers](#)

Question - 12:

Explain simple algorithm for bubble sort?

Ans:

```
void bubble(int x[],int n)
{
    int hold,j,pass;
    int switched=true;
    for(pass=0;pass<n-1;&&switched=true;pass++){
        switched=false;
        for(j=0;j<n-pass-1;j++){
            if(x[j]>x[j+1]){
                switched=true;
                hold=x[j];
                x[j]=x[j+1];
                x[j+1]=hold;
            }
        }
    }
}
```

[View All Answers](#)

Question - 13:

Tell me applications of linked lists and mostly used linked list?

Ans:

Used mainly to represent elements in a dynamic environment where it is added on an ad-hoc basis.

Especially in the cases where the total number of elements in the list cannot be pre-decided, linked lists are used. This does not lead to space insufficiency or space wastage as in case of arrays.

For eg. The no. of terms in a order-n polynomial varies greatly, using an array to store the co-efficients is an inefficient methods. If the array size is declared 100, a quadratic equation will use just 3 index and the rest 99 will be wasted. While for a sine or cosine series (from x to infinity) an overflow error might occur..!

[View All Answers](#)

Question - 14:

What are input function and output function in c language?

Ans:

printf,scanf,getch,getchar,getche,kbhit etc

[View All Answers](#)

Operating System Most Popular & Related Interview Guides

- 1 : [RTOS Interview Questions and Answers.](#)
- 2 : [Windows 7 Interview Questions and Answers.](#)
- 3 : [MAC Operating System Interview Questions and Answers.](#)
- 4 : [Disk Operating System \(DOS\) Interview Questions and Answers.](#)
- 5 : [Shell Scripting Interview Questions and Answers.](#)
- 6 : [Operating System \(OS\) Interview Questions and Answers.](#)
- 7 : [Solaris Interview Questions and Answers.](#)
- 8 : [Solaris Admin Interview Questions and Answers.](#)
- 9 : [VxWorks Interview Questions and Answers.](#)
- 10 : [OS Memory Management 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)