

# Frequently asked Data Structures Interview Questions

## Queues Data Structure – Interview Questions

---

### How is queue different from a stack?

---

The difference between stacks and queues is in removing. In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added.

### Why is a queue called a linear data structure?

---

Queue is said to be linear because its elements form a sequence or a linear list. Some other examples: Array, Linked List, Stacks

### Write a C program to implement a queue using two stacks.

---

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
    struct node *next;
};
void push(struct node** top, int data);
int pop(struct node** top);
struct queue
{
    struct node *stack1;
    struct node *stack2;
};

void enqueue(struct queue *q, int x)
{
    push(&q->stack1, x);
}

void dequeue(struct queue *q)
{
    int x;
    if (q->stack1 == NULL && q->stack2 == NULL) {
        printf("queue is empty");
        return;
    }
}
```

```

    }
    if (q->stack2 == NULL) {
        while (q->stack1 != NULL) {
            x = pop(&q->stack1);
            push(&q->stack2, x);
        }
    }
    x = pop(&q->stack2);
    printf("%d\n", x);
}

void push(struct node** top, int data)
{
    struct node* newnode = (struct node*) malloc(sizeof(struct node));
    if (newnode == NULL) {
        printf("Stack overflow \n");
        return;
    }
    newnode->data = data;
    newnode->next = (*top);
    (*top) = newnode;
}

int pop(struct node** top)
{
    int buf;
    struct node *t;
    if (*top == NULL) {
        printf("Stack underflow \n");
        return;
    }
    else {
        t = *top;
        buf = t->data;
        *top = t->next;
        free(t);
        return buf;
    }
}

void display(struct node *top1, struct node *top2)
{
    while (top1 != NULL) {
        printf("%d ", top1->data);
        top1 = top1->next;
    }
    while (top2 != NULL) {
        printf("%d ", top2->data);
        top2 = top2->next;
    }
}

int main()
{
    struct queue *q = (struct queue*) malloc(sizeof(struct queue));
    int f = 0, a;
    char ch = 'y';
    q->stack1 = NULL;
    q->stack2 = NULL;
    while (ch == 'y' || ch == 'Y') {
        printf("\n*****Enter your choice*****\n1.enqueue\n2.
scanf("%d", &f);
        switch(f) {
            case 1 : printf("Enter the element to be added to queue\n");
scanf("%d", &a);

```

```

        enqueue(q, a);
        break;
    case 2 : dequeue(q);
        break;
    case 3 : display(q->stack1, q->stack2);
        break;
    case 4 : exit(1);
        break;
    default : printf("invalid\n");
        break;
    }
}
}

```

## Stack – Data Structure Interview Questions

How do you implement a stack using a queue in a linked list?

```

#include<stdio.h>
#include<stdlib.h>

struct node
{
    int data;
    struct node * next;
};

struct queue
{
    struct node *rear;
    struct node *front;
};

void initial(struct queue *);
void qadd(struct queue *,int);
int qdel(struct queue *);
void dis(struct queue *);
void push(int);
void pop();

struct queue q1,q2;
int main()
{
    initial(&q1);
    initial(&q2);
    push(1);
    push(2);
    push(3);
    pop();
    printf("\nelements now are:\n");
    display(&q1);

    return 0;
}

void initial(struct queue *q)
{

```

```

    q->front=NULL;
    q->rear=NULL;
}

void qadd(struct queue *q,int n)
{
    struct node *tmp;
    tmp=(struct node *)malloc(sizeof(struct node));

    tmp->data=n;
    tmp->next=NULL;

    if(q->front==NULL)
    {
        q->rear=tmp;
        q->front=tmp;
        return;
    }

    q->rear->next=tmp;
    q->rear=tmp;
}

```

```

int qdel(struct queue *q)
{
    struct node *tmp;
    int itm;
    if(q->front==NULL)
    {
        printf("\nqueue is empty");
        return NULL;
    }

    //itm=q->front->data;
    tmp=q->front;
    itm=tmp->data;
    q->front=tmp->next;
    free(tmp);
    return itm;
}

```

```

void display(struct queue *q)
{
    struct node *tmp;
    tmp=q->front;
    while((tmp)!=NULL)
    {
        printf("\n%d", (tmp->data));
        tmp=tmp->next;
    }
    printf("\n");
}

```

```

void push(int val)
{
    struct queue tmp;
    int j;
    qadd(&q2, val);

    while(((&q1)->front)!=NULL)

```

```

    {
        j=qdel(&q1);
        qadd(&q2,j);
    }

    tmp=q1; //swap q1 and q2
    q1=q2;
    q2=tmp;

    printf("\nelements after pushing are:\n");
    display(&q1);
}

void pop()
{
    printf("\n element deleted is %d",qdel(&q1));
}

```

## Data Structures Common Interview Questions

---

If you are using C language to implement the heterogeneous linked list, what pointer type will you use?

---

The heterogeneous linked list contains different data types in its nodes and we need a link, pointer to connect them. It is not possible to use ordinary pointers for this. So we go for a void pointer. A void pointer is capable of storing a pointer to any type as it is a generic pointer type.

What factors determine the choice of data structure for a program?

---

As a programmer, we should consider the following factors. They are:

- The size of the data
- Time complexity
- The speed of data use
- The size of storage

What is the minimum number of queues needed to implement the priority queue?

---

Two. One queue is used for actual storing of data and another for storing priorities.

What are the data structures used to perform recursion?

---

Stack. Because of its LIFO (Last In First Out) property, it remembers its 'caller' so knows whom to return when the function has to return. Recursion makes use of system stack for storing the return addresses of the function

calls. Every recursive function has its equivalent iterative (non-recursive) functions. For these non-recursive functions also, system stack will be used.

### What are the methods available for storing sequential files?

Straight merging

Natural merging

Polyphase sort

Distribution of Initial runs.

### List out few of the applications that make use of Multilinked Structures?

Sparse matrix

Index generation.

### What is the type of the algorithm used in solving the 8 Queens problem?

Backtracking(stack application).

### In an AVL tree, at what condition the balancing is to be done?

If the 'Height factor' is greater than 1 or less than -1.

### What is the difference between stack and queue?

Stack follows LIFO (Last In First Out) principle whereas queue follows FIFO (First In Last Out) principle.

What are prefix and postfix notation for the expression  $A+B*C-D/E$ ?

Prefix notation for the given expression:  $-+A*BC/DE$

Postfix notation for the given expression:  $ABC*+DE/-$

when the overlapping and collision occur at the same time, then what is the size of the bucket?

---

One. If there is only one entry possible in the bucket, when the collision occurs, there is no way to accommodate the colliding value which results in the overlapping of values.

In RDBMS, what is the efficient data structure used in the internal storage representation?

---

B+ tree because in B+ tree, all the data is stored only in leaf nodes, that makes searching easier. This represents the records that shall be stored in leaf nodes.

Whether a Linked List is linear or Non-linear data structure?

---

According to Accessing strategies Linked list is a linear one.

According to Storage Linked List is a Non-linear one.

What do you mean by a free pool?

---

A Pool is a list consisting of unused memory cells which have its own pointers.

What is Asymptotic Analysis of an algorithm?

---

Asymptotic analysis of an algorithm refers to defining the mathematical foundation of its run-time performance. Using this analysis, we can conclude the best case, worst case, and average case scenario of an algorithm.

What are asymptotic notations?

---

Asymptotic analysis can provide three levels of mathematical binding of the execution time of an algorithm:

Best case by  $\Omega(n)$  notation.

worst case by  $O(n)$  notation.

average case by  $\Theta(n)$  notation.

Name some of the approaches to develop algorithms?

---

There are three commonly used approaches. They are:

there are three commonly used approaches. They are.

Greedy Approach – Finding a solution by choosing the next best option.

Divide and Conquer – Divide the problem to a minimum possible sub-problem and solve them independently.

Dynamic programming – Divide the problem to a minimum possible sub-problem and solve them combinedly.

### How many spanning trees can a graph have?

It basically depends on the connections of the graph. A complete undirected graph can have a maximum of  $n^{n-2}$  spanning trees, where  $n$  is the number of nodes.

### What is hashing?

Hashing is a technique to convert a range of key values into a range of indexes in an array. By using hash tables, we can create an associative data storage where data index can be found by providing its key value.

### When does a Spanning Tree could be called as Minimum Spanning Tree?

In a Weighted Graph, a Minimum Spanning Tree is a spanning tree that has the minimum weight among all spanning trees of the same graph.

### How to find middle element of linked list in one pass?

To find the middle element of the middle element of linked list in one pass, we need to maintain two pointers, one increment at each node while other increments after two nodes at a time. Through this arrangement, when the first pointer reaches the end, the second pointer will point to the middle of the linked list.

### How to find if a linked list has a loop?

While finding middle element of linked list in one pass, there may be a chance that one node will be pointed by the two pointers, in this case, we can say that the linked list has a loop.

### What are the different techniques for making hash functions?

Techniques for making hash functions are:

- Truncation method



- Mid square method
- Folding method
- Division method

### What is Huffman's algorithm?

It is used in creating extended binary trees that have minimum weighted path lengths from the given weights. It makes use of a table that contains the frequency of occurrence for each data element.

### What is a dequeue?

A dequeue is a double-ended queue. The elements can be inserted or removed into this queue from both the ends.

### Can a stack be described as a pointer?

A stack can be represented as a pointer. Stack has a head pointer which points to the top of the stack. Stack operations are performed using the head pointer. Hence, the stack can be described as a pointer.

### What is space complexity and time complexity?

The space complexity of a program is the amount of memory consumed by the algorithm until it completes its execution. An efficient algorithm takes space as small as possible.

The time complexity is the amount of time required by an algorithm to execute. It allows comparing the algorithm to check which one is the efficient one.

### What is the difference between Binary Tree and Binary Search Tree?

Binary Tree: In a binary tree, each node can have a maximum of 2 child nodes, and there is no ordering in terms of how the nodes are organized in the binary tree. Nodes that do not have any child nodes are called leaf nodes of the binary tree.

Binary Search Tree: Binary Search Tree is essentially a binary tree, in terms of how many child nodes a node in the binary search tree can possibly have.

The important difference between a Binary Tree and a Binary Search Tree: In a binary search tree there is a relative ordering in how the nodes are organized, while there is nothing of that sort in a binary tree. In Binary search tree, all the nodes to the left of a node have values less the value of the node, and all the nodes to the right of a node have values greater than the value of the node.

## What is queuing simulation?

Queuing simulation as a method is used to analyze how systems with limited resources distribute those resources to the elements waiting to be served, while waiting elements may exhibit discrete variability in demand, i.e. arrival times and require discrete processing time.

## List out the applications of minimum spanning tree?

- Telephone exchanges
- Networking of computers in the lab for minimizing the length of wire.
- It provides a reasonable way for clustering points in space into natural groups.

## List out the applications queues?

Two major applications of a queue are:

QueueSimulation: It is a modeling activity used to generate statistics about the performance of the queue.

Categorization: Queues are used to categorize data into different groups without losing the order of the original data.

## List out the common operations that are associated with lists?

- Insertion
- Deletion
- Retrieval
- Traversal

## What is the maximum and minimum height of a binary tree having N nodes?

The maximum height of the binary tree,  $H_{max}=N$

The minimum height of a binary tree,  $H_{min}=\lceil \log_2 N \rceil + 1$

## What are the three approaches for inserting data into general trees?

The three approaches are:

- FIFO(First In First Out)
- LIFO(Last In First Out)

- Key sequenced

### What are the properties that make a binary tree as a binary search tree(BST)?

The properties of are:

- All items on the left subtrees are less than the root.
- All items in the right subtree are greater than the root or equal to the root
- Each subtree is itself a binary search tree.

### Which traversal in the binary search tree will produce a sequenced list?

In order traversal in the binary search tree(BST) produces a sequential list. This traversal will give a sequential list in ascending order.

### List out the applications of a heap?

The common applications of heaps are selection, priority queue, and sorting.

### What are the storage structures to represent graphs?

The storage structures to represent graphs are:

Adjacency matrix

Adjacency list

### What are the standard traversals in the graphs?

The two standard graph traversals: depth-first and breadth-first.

In the depth-first traversal, all of a node's descendants are processed before moving to an adjacent node.

In the breadth-first traversal, all of the adjacent vertices are processed before processing the descendants of a vertex.

### What are the major data structures used in the given areas: RDBMS, Network data model, and hierarchical data model?

RDBMS –arrays

Network data model –graph

Hierarchical data model –trees

Which notations are used in the evaluation of arithmetic expressions using prefix and postfix forms?

---

The notations used are Polish and Reverse notation.

When is a binary search best applied?

---

A binary search algorithm that is best applied to search a list when the elements are already in the order or sorted.

In what areas the structures are used?

---

Data structures are used in every aspect where data is involved. The following areas use algorithms of data structures most efficiently. They are:

- Numerical analysis
- Artificial intelligence
- Database management structures
- Compiler designing
- Operating systems
- Networks

How does dynamic memory allocation help in managing data?

---

Dynamic memory allocation can combine separately allocated structured blocks to form composite structures that expand and contract as needed.

What is the primary advantage of a linked list?

---

The main advantage of a linked list is that it can be modified easily. The modification of a linked list works regardless of how many elements are in the list.

What is the advantage of a heap over stack?

---

The heap is more flexible than the stack. That's because memory space for the heap can be dynamically allocated and de-allocated as needed.

### What is an AVL tree?

An AVL tree is a search tree in which the height of the subtree differs by no more than one. It is otherwise called as a balanced binary tree. For AVL tree the balancing factor will be `-1, 0, 1` only.

### What are the properties associated with the heaps?

The properties are:

The tree is complete or nearly complete

The key value of each node is greater than or equal to the key value in each of its descendants.

### What is the efficiency of Insertion Sort, Selection Sort, and Bubble Sort?

The efficiency of the insertion sort, selection sort and bubble sort in big-O notation is  $O(n^2)$ .

### What are the variations in the sequential search? Explain about them.

There are two variations in a sequential search. They are

Sentinel search

Probability search

Sentinel search: With this method, the condition ending the search is reduced to only one by artificially inserting the target at the end of the list.

Probability search: in this method, the list is ordered with the most probable elements at the beginning of the list and the least probable at the end.

### Using which data structure format, the hash tables are storing values?

Arrays is used to store the content in hashtables.

## How many stacks are required to implement a queue?

---

Two stacks are required to implement a queue.

For enqueue: take two stacks, say s1 and s2; perform push on s1.

For dequeue: if s2 is empty, pop all the elements from s1 and push it to s2. The last element you popped from s1 is an element to be dequeued. If s2 is not empty, then pop the top element in it.

## How is an array different from a linked list?

---

A The size of the arrays is fixed, Linked Lists are Dynamic in size.

Inserting and deleting a new element in an array of elements is expensive, Whereas both insertion and deletion can easily be done in Linked Lists.

Random access is not allowed on Linked Listed.

Extra memory space for a pointer is required with each element of the Linked list.

Arrays have better cache locality that can make a pretty big difference in performance.

## What data structure is used to implementing BFS(Breadth First Traversal) and DFS(Depth First Traversal)?

---

A queue is used to implement BFS.

A stack is used to implement DFS.

## What is LRU caching scheme in data structures?

---

The LRU caching scheme is to remove the least recently used frame when the cache is full and a new page is referenced which is not there in the cache when we have given information about total pages and cache size.

## What are the two data structures that are used to implement LRU caching?

---

A Queue is implemented using a doubly linked list. The maximum size of the queue will be equal to the total number of frames available (cache size). The most recently used pages will be near the front end and least recently pages will be near the rear end.

A Hash with page number as key and address of the corresponding queue node as value.

### List out different types of loops with its efficiencies in data structures?

Following table demonstrates the efficiency trends for different data structures.

Data Structure	Efficiency	
Logarithmic loop	$f(n)=\log n$	
Linear loop	$f(n)=n$	
Linear logarithmic loop	$f(n)=n(\log n)$	
Quadratic loop	$f(n)=n^2$	
Dependent quadratic loop	$f(n)=n(n+1)/2$	

### What are the two approaches that are used to write repetitive algorithms?

Iteration and recursion are the two repetitive algorithms. Recursion is a repetitive process in which an algorithm calls itself. Iteration algorithm is generally used with loops.

### What is the general rule for designing a recursive algorithm?

Determine the base case.

Determine the general case

Combine the base case and general case in the algorithm

### What are the two features that most affect the performance of queues?

The most important features are its arrival rate and the service time.

The rate at which the customer arrives into the queue for service is known as arrival rate.

The average time required to complete the processing of customer request is known as service time.

## Data Structures

◆ [Data Structures Tutorial](#)

◆ [Data Structure - Queue](#)

◆ [Data Structure - Stack](#)

◆ [Data Structure - Tree](#)

◆ [Data Structures - Sorting Algorithms](#)

⇒ [Data Structures Interview Questions](#)