Nagarro

Nagarro-Software is an organization that provides business consulting and software development services. It was established in 1996 in Munich, Germany, and has its headquarters in Gurgaon, India. Nagarro Software has operational centers in North America, Europe, and Asia, and has operational locations in those regions. From 2011 to 2016, Nagarro Software merged with many companies, resulting in a division with over 3,000 employees. As an example of the firm's dedication to social good, it sponsors Raahgiri Day, a bicycle-sharing program, offering free parking for employees, as well as other activities.

Nagarro Recruitment Process

1. Interview Process

Nagarro selects the candidates for both fresher and experienced roles in 4 different processes. Nagarro hires freshers from both on-campus, off-campus, and employee referrals. Experienced candidates can get the interview call by applying from the career portal or Employee Referrals.

- Round 1 - Online Test (Aptitude + Technical).
- Round 2 - Online Coding Test.
- Round 3 - Technical Interview.
- Round 4 - HR Interview.

Note: Every round is an elimination round. You will get the call for the next round only if you pass the current round.

2. Interview Rounds

- **Round 1: Online Test (Aptitude + Technical)-** This round is the online test consisting of (**Aptitude and Technical Questions**) and will be conducted on the **Mettl Platform**. This will be an **MCQ Based** round and it will consist of a maximum of 40 questions with a time limit of **60 minutes**. This round is divided into two sections Aptitude and Technical Section. Those are -
  - **Aptitude Section -** In this section, there will be 15 questions. That will be based on the topics like Verbal Reasoning, Numerical Reasoning, Logical Reasoning, and General Aptitude tests.
  - **Technical Section - I**n this section, there will be 25 technical questions. And the questions will mostly be based on Data Structures and Algorithms. In some cases, the company will also ask for a maximum of 5 code snippets in which you need to identify errors or predict the output.
- **Round 2: Online Coding Test** - This round is an online coding test consisting of **5 Data Structures and Algorithm based coding questions**. The difficulty level of the question will be medium to hard. Nagarro mostly asks questions from the Tree and Graph. In some cases, it can also be from the String and Array-based questions, and approaches are mostly on the

Dynamic Programming. The duration of the test will be of **3 hours** and will be conducted on the **Mettl Platform**.

- **Round 3: Technical Interview -** This round is the Technical Interview Round. The interview duration will be of 1 hour and the questions will be asked on various topics related to the computer subjects and followed by 1 or 2 coding questions and SQL Query. This depends on the interviewer and what questions they will ask. But in general, the topics from which the questions will be asked are - Data Structures, Analysis of Algorithms, Computer Networks, Operating systems, Database Management systems, Object-Oriented Programming, Programming Language from resume (Preferred Java), coding questions, and SQL Query. This round will be conducted on Microsoft Teams.
- **Round 4: HR Interview -** This round is an HR Interview Round. This will be your general talk with HR. And the behavioural questions will be asked. Most often, this is a non-elimination round for freshers but you need to take care of the answers. They usually check your interest in the company and how excited you are to work for the Nagarro. But if you are an experienced candidate and applying for the senior position then they check the background of your previous work and many other things.

Note - These rounds will be common for both Fresher and Experienced Candidates applying for the role of associate software developer, software developer, and senior software developer.

Nagarro Internship

After you are selected for the role of Associate Software Developer or Software developer as a fresher, then you get an internship offer from Nagarro. During the internship, the company will train you on the domain and will provide you with a stipend of **₹19,000/month** for an Associate Software Developer and **₹25,000/month** for a Software Developer.

Nagarro HR Interview Questions

The interviewer will start by asking the basic questions and then will ask some questions to check your communication skills and Interest in the company for freshers. And for an experienced candidate, the questions can also be from your previous work experience. The questions can be -

- **Tell Us About Yourself**

Self-introductions are awkward, but they are a necessity. You only have a few seconds to introduce yourself and sell yourself. Self-introductions can be tricky, but they also don't have to be. Start by telling the interviewer a little bit about your childhood and family, your education? Followed by your work experience (in case you have any). Be genuine, but don't get too in-depth or too delicate. Then, tell the interviewer about a specific accomplishment that shows off a skill you've mastered.

- **Why Do You Want to Work for Nagarro?**

  You don’t have to explain why you want the job. But you do have to explain why you want the job in a specific role or field. Tell the interviewer about the job description that matches your skill and the principles that are followed by the company, and how that suits you.

- **Why Should We Hire You?**

  This question is a good opportunity for you to explain your skills and accomplishments in a way that shows off your personality and why you are the best choice for the job. If you get this question wrong, you’ll be in trouble. Employers don’t have time to listen to long-winded answers. You have just seconds or milliseconds to impress them. Employers don’t want to hear about how great you are. They want to know why you’re the best choice for the job. Make sure that you have a solid answer for why you should be hired, and employers will like you even more.

- **Sum up your previous work experience.**

  This is a great question because it gives you a chance to highlight your skills and accomplishments from the past. You can also use this question to explain how your skills have evolved and how your skills will benefit the company.

- **What is your greatest strength?**

  If an interviewer asks you this question, it’s likely because he or she wonders what your greatest strength is. A skill that’s perfect for one job may be completely irrelevant for another role. Therefore, the interviewer is trying to unearth what makes you special, both in your skills and personality. The interviewer wants to find out what makes you a natural fit for the job, even if your skills don’t match perfectly. When you’re asked what your greatest strength is, you can either quickly mention something specific or dig a little deeper by explaining how you’ve flourished in a particular environment. If the interviewer asks about your strengths, this is your chance to shine.

- **Why do you want this job?**

  This is a great question to ask because it gives the interviewer an insight into your motivation. Why do you want this job? If a lot of people want the job, it’s probably because it offers a lot of money. So why do you want the job if it doesn’t provide enough pay?

- **Tell me about a time when you demonstrated loyalty and teamwork.**

  Employers want to know how you deal with conflicts and challenges with loyalty, teamwork, and discipline. If you’re a person who’s satisfied with mediocrity and never pushes harder than your average, you’re not likely to stand out. If you can’t answer this question and explain how you work with others, then you'll have a whole lot of explaining to do when you're not selected for a job.

- **Tell me about a time when you had to work with someone difficult.**

  You don't have to be in a high-pressure situation to work with someone difficult. Every job has its challenges and challenges. Once you have to work with a difficult person and handle it well, employers will wonder how you handle other challenges with ease.

- **Tell me about a time when you took initiative and showed courage.**

  Initiative and courage are important qualities that employers want to see in every candidate. If you don't show initiative and courage in your past jobs, you might get noticed for the wrong reasons. Explain a time when you took initiative and showed courage. It doesn't necessarily have to be a large-scale move. It can be something as small as calling your supervisor to ask how they want to be organized.

- **What do you like to do for fun?**

  This is a question that is often asked to get a sense of what your lifestyle is like and if it follows the principles of the company. It can come up in a job interview, or in an informal conversation. A good way to answer this question is to try to tie your hobbies back to the job. Maybe hiking is something you do in your free time, or maybe something from your last job is more relevant. Try to tie your hobbies back to the job, and explain how these activities help you to relax, or relieve stress.

- **Why do you want to leave your current job?**

  This is a question that is often asked in the hiring process, but it can also come up in a job interview. It's a good question to ask to see if you're interested in the job for the right reasons. A good way to answer this question is to try to downplay it. Simply say that you're interested in the job, or want to learn more about it, or that you want to change career paths. You can try to downplay the question by saying that you're not sure why you want to leave your current job, or that it's too hard to explain.

Nagarro Aptitude Test Questions

The aptitude questions can be from different topics like numerical ability tests, verbal reasoning, etc. These are basic questions that you can practice by solving more questions from the topics. The questions that can be asked are like -

1. A village's population increased by 5% from 2007 to 2008 and by 25% from 2005 to 2009. From what amount, proportionate to 480, did the population grow?

a) 630
b) 610
c) 640
d) 620

Answer: a) 630

Explanation- Population in 2007 is : 480
Population in 2008 = 1.05 × 480 = 504
Population in 2009 = 1.25 × 504 = 630

2. Hiren allocated 45% of his monthly pay to paying bills and rent in March. He then invested 60% of his extra cash into the PPF scheme and kept the remaining 20%. He deposited $15,400 into his bank account. If his salary increased by 10% in April, what was it?

a) ₹ 80,000
b) ₹ 59,000
c) ₹ 1,10,000
d) ₹ 77,000

Answer: d) ₹ 77,000

Explanation- Let the initial salary of Hiren be 'x'. Then:
(0.55 × 0.4) x = 15400 => x = 70000
After increment, Hiren's salary = 1.1 × 70000 = 77000

3. 120 litres of milk were mixed with 24 litres of water in Jar A. 12 litres of the resulting mixture were withdrawn and 3 litres of water were added. How many litres of the new mixture will be produced if 27 litres of the resulting substance are removed?

a) 10 litres
b) 20 litres
c) 15 litres
d) 30 litres

Answer: b) 20 litres

Explanation - Milk Ratio : water in Jar A = 120 : 24 = 5 : 1
12 liters of this mixture is taken out => milk = 5/6 × 12 = 10 liters and water = 2 liters are taken out
3 liters of water more added = 24 – 2 + 3 = 25 liters => new ratio of milk : water
= 110 : 25 = 22 : 5

Now, 27 liters of this mixture is again taken out =>
water taken out = 5/27 × 27 = 5 liters
water left = 25 – 5 = 20 liters

Ques- A boat can travel 30 kilometres downstream at 15 kilometres per hour and return upstream in 4 hours 30 minutes. What is the downstream speed of the boat?

a) 10 km/h
b) 6 km/h
c) 5 km/h
d) 4 km/h

Answer: c) 5 km/h

Explanation - Speed of boat = 15 km/h
Let the speed of the stream be 'x'
Given: 30/(15+x) + 30/(15-x) = 9/2
By Solving this, we get, x = 5 km/h

Nagarro Technical Test Questions

In the Technical test, the questions can be asked mostly from Data Structures and Algorithms, along with some computer fundamentals. The questions are like this -

1. What are the levels of data structure implementation?

a) Abstract Level.
b) Implementation Level.
c) Application Level.
d) All of the above.

Answer: d) All of these

2. PPP identifies the Network layer protocol using what protocol?

a) NCP
b) ISDN
c) HDLC
d) LCP

Answer: a) NCP

3. What is missing from the Abstract Data Type description?

a) Data.
b) Operation.
c) Both.
d) None.

Answer: d) None

4. What kind of structure enables both deleting data items and adding them at the back?

a) Stack.
b) Queue.
c) Binary Search Tree.
d) Dequeue.

Answer: a) Stack

5. From the given options. Find the operating system that is not a real-time operating system.

a) VX Works.
b) RT Linux.
c) Palm OS.
d) Windows CE.

Answer: c) Palm DS

Nagarro Technical Interview Questions: Fresher's and Experienced

1. Explain Stack as Data Structure. Also, explain what are the real-time applications of Stack.

The Stack is a collection of data that are stored one on top of another. Stacks can be used to make sure that data is kept in order and has a constant backup in case of an incident. It can also be used to optimize the organization of your data by storing it in the most efficient way possible.

For example, if you have a lot of data about your customers that you would like to organize, you could store all your customer records on one big stack. Then, when you need information about a particular customer, you can just pull out their record from the stack. This makes finding information much easier.

Stack data structures are widely used in programming. They are the most commonly used data structure in the world of computers. This is because it simplifies the process of storing and retrieving data.

It is also useful for storing several values of the same type. For example, you can use a stack to store passwords. The stack will keep track of which password you have already entered. When you enter a new password, it will be pushed onto the stack. When it reaches the top, it is removed from the stack, and you will be prompted to enter a new password.

The other main advantage of using a stack is that it makes it easier to build algorithms that perform array operations. This is because there are no arrays involved. Instead, it takes just one value at a time and pushes it onto the top of the stack until it reaches the bottom again. This makes things much simpler than using an array when performing certain operations like addition, subtraction, and comparing items in an array.

2. Explain something about PL/SQL.

- **PL/SQL** is a procedural programming language that runs on top of the SQL database. It has its syntax, but it can also be used in conjunction with SQL to access the full range of features offered by the database. PL/SQL is often used for business applications because it is capable of performing more complicated tasks than SQL alone. Also, because it runs on top of the SQL database, it can take advantage of the full range of features that the database offers. This includes data integrity checks and advanced security features such as user authentication and authorization control.
- PL/SQL is a high-level language that is easy to learn and use. It makes it possible to write complex, flexible applications that are able to perform complex tasks with ease.
- One key advantage of PL/SQL is its ability to run on top of a relational database, giving it access to all the data integrity and security features that are available in SQL. Another advantage is its ability to work with other languages, such as HTML and Cascading Style Sheets (CSS). This means that you can easily integrate functionality from other applications into your PL/SQL program without having to rewrite any code.

3. What are Data Segments? And how does this relate to system performance?

A data segment is a portion of the disk space that is allocated to store user data. The disk space can be divided into segments, which are containers for storing different types of data. There are three types of segments: Physical, Leachable, and Non-Leachable.

- Physical segments store your data in the hard drive itself.
- Leachable segments are those that contain data that you have permission to access.
- Non-leachable segments contain information that you do not have permission to access, such as system files and cached data.

It affects the system performance as -

The amount of time that it takes for your computer to respond when you press a key on your keyboard depends on the speed at data transfer from your hard drive to the processor. If there is a bottleneck in one area, it can affect performance in other areas. Lack of free storage space can also cause system slowdowns because the operating system has to move more data around than it needs to. To keep things running smoothly, make sure you keep sufficient free space on all drives and keep your system clean by regularly deleting unnecessary files.

4. What are Codd Rules? Explain Each of them.

Dr Edgar F. Codd established twelve rules for databases that adhered to the Relational Model after conducting extensive research on it. He said that a database must follow these rules in order to be considered a true relational database. All databases that rely on relational functionality to store data should follow these basic rules.

- **Information Rule**
  Every piece of data stored in a database must be represented as a value of a table cell. It is imperative to keep everything in a table format.
- **Guaranteed Access Rule**
  Every data element/value that is present in the database can be accessed logically using a table name, primary key (row_value), and attribute name (column_value). No pointers or other techniques may be used to access the data.
- **Systematic Treatment of NULL Values**
  It is very important to treat the NULL values in a database systematically and uniformly. This is because a NULL may be thought of as being missing data, unconfirmed data, or nonstandard data.
- **Active Online Catalog**
  An online catalog of the database's structure must be maintained, known as a data dictionary. Users may query the catalog using the same query language used for accessing the database itself.
- **Comprehensive Data Sub-Language Rule**
  A database must be only accessible using a language that has the linear syntax, and it must support data definition, data manipulation, and transaction management operations, and can be used directly or through some application. This language may be used directly or through some application. If the database provides access to data without the help of this language, it is considered a violation.
- **View Updating Rule**
  Every aspect of a database that may have been altered, must also be accessible to the system.
- **High-Level Insert, Update and Delete Rule**
  It is necessary for a database to support high-level insertion, updating, and deleting of data records. The database doesn't need to support only a single row of data, that is, it must also support the union, intersection, and negation operations to yield sets of data records.
- **Physical Data Independence**
  Accessing a database should not depend on the physical structure of the database. Any change in the physical structure of a database should not affect how the data is accessed by external applications.

- **Logical Data Independence**
  A database must maintain logical data that is independent of the way it is used by applications. When two tables are combined or one is split into two new tables, there should be no effect on the applications. Because this is such a difficult objective to achieve, this rule is usually difficult to comply with.
- **Integrity Independence**
  It is critical for a database that is used by an application to be independent of the application and interface. Integrity constraints of the database can be altered without modifying the application. This is because the front-end application and its interface are not tied to the database.
- **Distribution Independence**
  The data should always appear to be located at a single location to end-users. This fundamental design principle has been built upon the notion that users should never be able to see that data is distributed across various locations.
- **Non-Subversion Rule**
  A system's interface must not be able to subvert the system and bypass security and integrity constraints if the interface provides low-level access to records.

5. What is Database Trigger?

A database trigger is a SQL statement that is run when you insert, update, or delete a row in your database. It allows you to execute code before or after the data is inserted into your database. This can be useful if you want to check for certain conditions before inserting data into your database, such as making sure the value of one field does not already exist. It also has other uses, such as checking if a user is logged in before allowing them to modify a record or deleting an existing record. A database trigger can be written in any programming language, although some languages are better suited than others. There are two types of triggers: Stored and User-Defined.

- Stored triggers execute the code when a row is inserted (INSERT), updated (UPDATE), or deleted (DELETE).
- User-defined triggers can have custom logic built on top of them using VALUES.

6. What is the difference between Rename and Alias?

Rename is a task that changes the name of an object. Alias is a task that creates a symbolic reference for an object. Creating an alias means that you can refer to the object by a different name, but this does not change the underlying identity of the object or its value.

The main difference between aliasing and renaming is that renaming changes the value of an object. Aliasing only changes how the object is identified.

To understand how aliases and renaming differ, let’s consider an example. Suppose that you have an integer value that represents a person’s age in years. You can represent this value as 3, but it would be more useful if you could also represent the person’s age as 82. With renaming, you can accomplish this by calling the value “Bob”. Now, “Bob” is a name, but it doesn’t change the number 3 into 82. With renaming, you can change the value of “Bob” from 3 to 82.

7. What is an abstract class? And what is the advantage and disadvantage of using abstract classes?

An abstract class is a class that is defined without any implementation. Abstract classes can be used to provide a common base for other classes or to define an interface that can be implemented by other classes. Abstract classes are not directly usable by programs. However, they can be useful in defining interfaces that other classes can implement and use. Abstract classes can also be used to provide shared functionality that other classes need access to but do not need to implement completely.

- The main advantage of using an abstract class is that it can allow programmers to write code that only needs to access the abstract methods defined in the class, while other code in the program can use the methods in the class if they are needed. This allows programmers to write simpler code that does not need to worry about all the details of how the abstract methods will work. If another class or subclass needs access to an abstract method, it simply needs to implement the abstract method and call it as required, without having to worry about exactly how it will work when it is called.
- The main disadvantage of using an abstract class is that it cannot be used for any actual implementation. In order for an abstract class to be used, all its methods must be defined and implemented before it can be used by any other code. If any of these methods is left undefined or unimplemented, then the program will fail when trying to use the class.

8. What are Artificial intelligence and Machine Learning? What is the difference between them?

Artificial intelligence (AI) is the ability of a machine to behave like a human. AI is a field in computer science that deals with the creation of machines that can perform tasks normally requiring human intelligence. AI has many potential uses, including automated planning, speech recognition, and translation. AI has been used in both government and industry for many years.

As AI becomes more sophisticated, more applications are emerging. AI can be broken into two main categories: Supervised learning and Unsupervised learning.

- In supervised learning, an algorithm trains on large amounts of data and learns from this data.
- Unsupervised learning does not require a training set; instead, it learns from unlabeled data and then labels the data using its own parameters.

Machine learning is different from AI because it focuses on algorithms that learn how to detect patterns in data without being explicitly programmed to do so. It is also different from AI in that it does not involve creating a machine that behaves as a human would. Rather, it involves creating algorithms that can extract useful information from data sets.

9. Explain about TCP/IP Protocol?

TCP/IP is the networking protocol that allows computers to communicate with each other over a network. It is the foundation of modern internet communication. TCP/IP is an acronym for

Transmission Control Protocol (TCP) and Internet Protocol (IP). Together they form the TCP/IP protocol suite, which is used to route information from one location to another over the internet. These two works together in order to ensure that data is sent from one computer to another without any problems. TCP/IP is the most widely used protocol today because it allows devices on a network to communicate with each other regardless of their location. It's also responsible for routing data between networks, so it's important for any device that needs to send or receive data across networks. This includes computers, smartphones, tablets, and even smart appliances like security cameras and baby monitors.

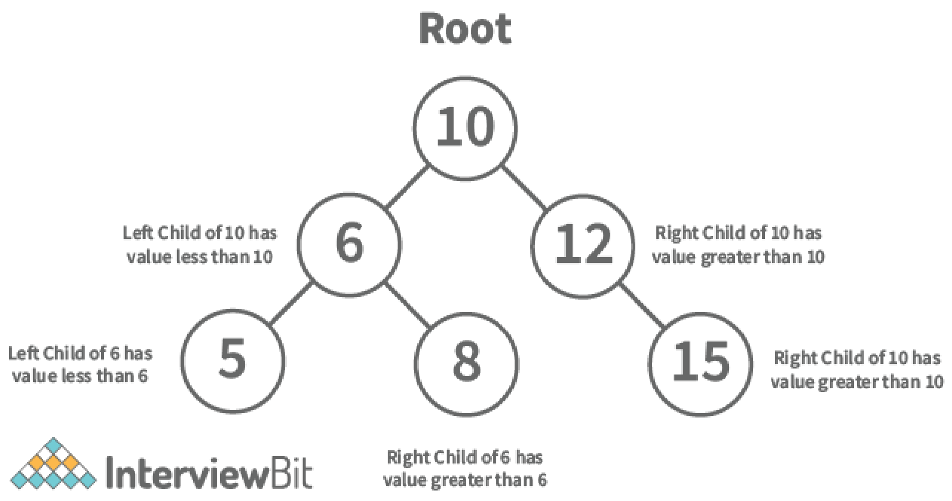
10. Explain Binary Search Trees?

A binary search tree (also known as Binary Search Tree or BST) is a data structure that organizes data into two different types of nodes, which are leaves and internal nodes. Each node has a key value that uniquely identifies it. The purpose of a binary search tree is to organize data so that it can be accessed efficiently by keywords. In order to access the data, you must always start at the root node and then follow the path down the tree until you reach an internal node that contains the value you're looking for. Each internal node will have one or more child nodes, which will also have keys that identify them. As you continue to traverse the tree, you will keep on finding higher-level nodes (which are more deeply nested) until you eventually reach a leaf node that contains your desired data.

Binary Search tree inherits the property of Binary Tress. But in addition to this, it also has its own property that -

- For every root node, values less than the root node will exist on the right subtree.
- For every root node, values greater than the root node will exist in the left subtree.

And this follows for every individual node in the tree. Consider the below image for a better understanding.


Root
10
6
12
5
8
15
Left Child of 10 has value less than 10
Right Child of 10 has value greater than 10
Left Child of 6 has value less than 6
Right Child of 10 has value greater than 10
Right Child of 6 has value greater than 6
InterviewBit

In addition to organizing data in a structured way, binary search trees also provide additional functionality such as efficient searching and sorting because of the two properties of the binary search tree.

Binary search trees are commonly used in computer science applications such as databases and networking systems because they allow for fast retrieval of the most important information from large sets of data.

11. What is Cache Memory?

Cache memory serves a similar purpose as RAM in a computer. Cache memory is temporary storage that can be used to store frequently accessed data.

Being able to quickly access data can improve the performance of a system by reducing the amount of time it takes for a processor to access the data. Having more cache memory also results in faster processing times.

Cache memory is typically much faster than RAM and can be used to temporarily store frequently accessed data like application data, metadata, or result data. It's commonly referred to as "system cache" because it tends to be located on the same die (or integrated circuit) as the processor itself.

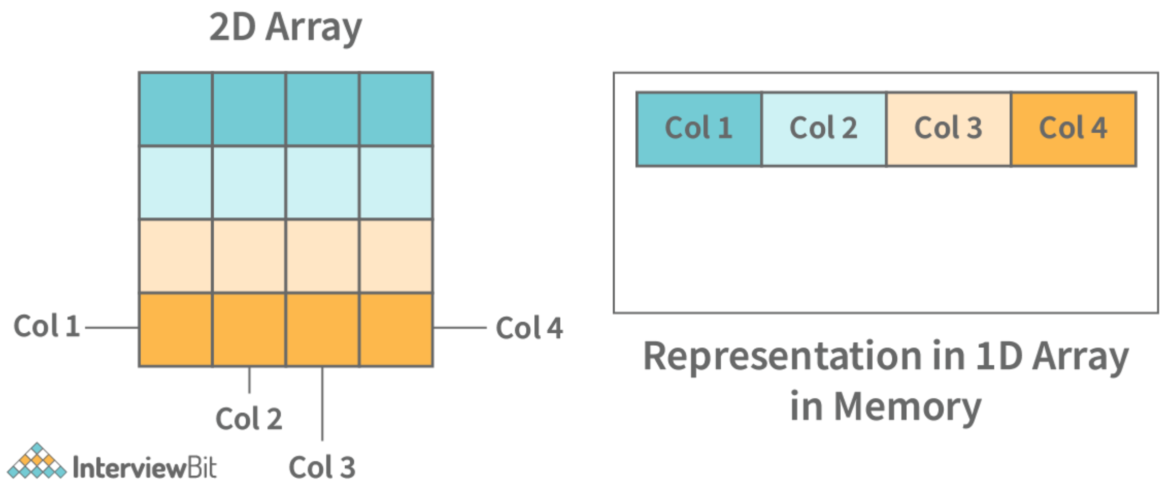
12. What is a Critical Section in Operating System?

A critical section is a section of code in an operating system where the execution cannot proceed until all threads reach the section. If any thread attempts to access the critical section before the other threads are finished, it will be blocked until all threads have reached the section. As a result, critical sections can be used to ensure that no thread can interrupt another thread's execution.

13. Explain row-major order and column-major Order.

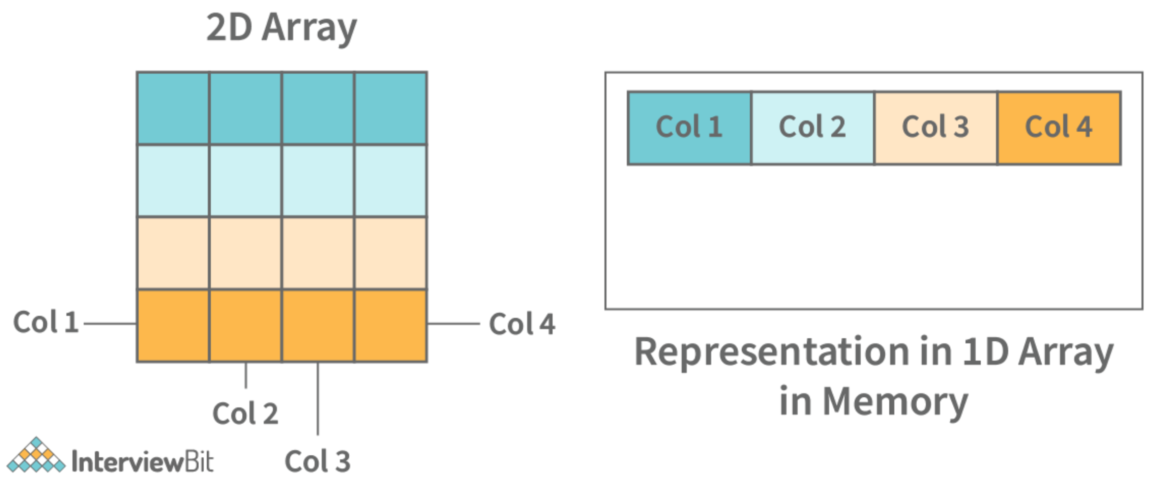
The data from two-dimensional arrays are in a contiguous fashion. So how these data are to be stored in the memory are derived by the row-major order and the column-major order.

- In row-major order, first, the entire row will be stored one after another in a continuous manner. For example - Consider the below image.



In the above image, we can see that the first row is filled first and then the second row, and so on.

- In column-major, the data in the two-dimensional matrix is represented and stored in the memory column-wise. ***For Example -*** Consider the below image.



In the above image, we can see that the first column is filled first and then the second column, and so on.

14. What are public and private members of a class?

In object-oriented programming (OOP), there are two types of members that can be accessed by any type: private and public. Private members are used for storing data, while public members are used for accessing that data. Private members are inaccessible outside of the class itself, while public members can be used by any other class in the same program. This means that any variables declared as public can be accessed by another class, but private variables cannot be accessed at all.

Public and private members are also referred to as "class variables" and "instance variables."

A good example of a private member is a variable that is defined inside a class but not accessible from outside the class. This is done to keep sensitive information away from other classes in the same project.

Another example of a private member is a function that's only available within a particular class. The purpose of this function is to provide specialized functionality for a particular class or subclass. The best way to understand private and public members are by looking at their context and purpose.

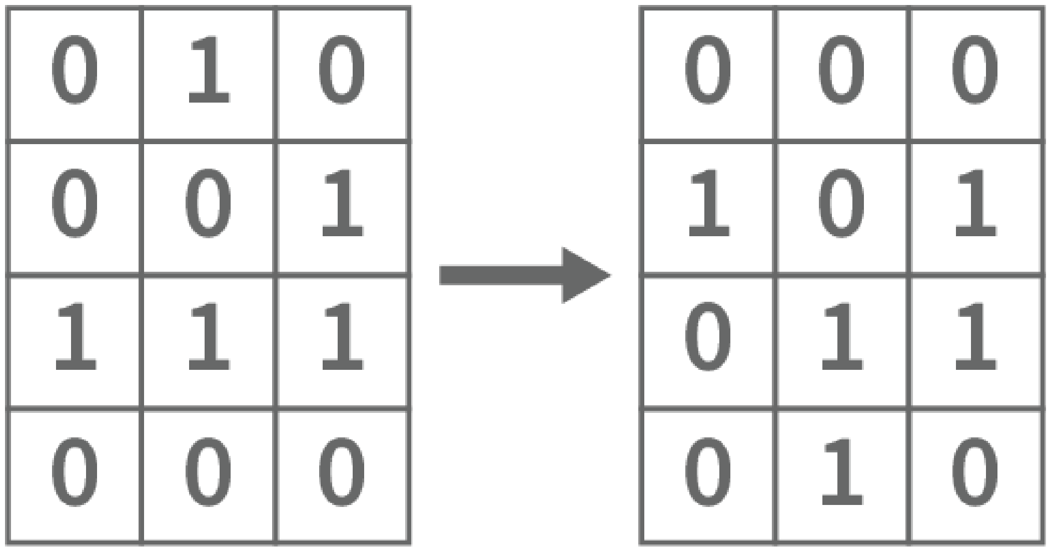
15. What are serialization and deserialization OOPs?

When an object is passed around between different parts of an application, it needs to be serialized into a standard format so that it can be transmitted safely across the network without accidentally corrupting it or causing other problems.

- Serialization is the process of converting an object into a compact representation that can be stored and transported reliably. A serializer takes some data of type Object and converts it into another data type: String (or ByteArray) or ArrayList etc. The rest of OOPs is about encapsulation and abstraction - separating data from its internal representation so the user doesn't have direct access or knowledge about its implementation details.
- Deserialization on the other hand is the reverse process of serialization. It is the process of converting the serialized data stream object into the actual instance of the class. Deserialization is often used to convert data back from our storage systems when we need it again.

16. An m x n grid of cells is used, where each cell has either a life (as 1) or dead (as 0) state initially. When each cell interacts with its eight neighbors (horizontal, vertical, and diagonal), the following four rules govern their behavior.

- **When a live cell has fewer than two live neighbors, it dies because of being under-populated.**
- **A cell with two or three living neighbors survives to the next generation.**
- **An overcrowded 1 live cell kills off any cells with more than three live neighbors.**
- **Whenever a dead cell comes across three live cells, it becomes alive again by reproduction.**
- **The current state is computed by applying the above rules to every cell in the m x n grid board in succession. Write a program to return to the next state.**



One of the approaches is that we can use the auxiliary grid with the old value and then we update the grid using the above 4 rules. This takes O(m*n) time and also O(m*n) extra space. But we can solve this problem in-place by keeping a track of the previous value of the grid with some unique value for each state. The special number that we considered to count the sides with 1.

- 1 changes to 0 = -2
- 0 changes to 0 = 0
- 1 changes 1 = 1
- 0 changes to 1 = 2

With this above value, we can proceed with the solution. The below solution will be like this -

```
  //Method that updates the value of the grid.
   public void gameOfLife(int[][] board) {
        /*
            A special number that we considered to count the sides with 1
            1 -> 0 = -2
            0 -> 0 = 0
            1 -> 1 = 1
            0 -> 1 = 2
        */
        int m = board.length;
        int n = board[0].length;

        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                int count = 0;
                //Counting the dead cell in the neighbor
                if(i-1 >= 0){
                    if((j-1 >= 0) && ((board[i-1][j-1] == 1) || (board[i-
1][j-1] == -2)))
                        count++;
                    if((board[i-1][j] == 1) || (board[i-1][j] == -2))
                        count++;
                    if((j+1 < n) && ((board[i-1][j+1] == 1) || (board[i-
1][j+1] == -2)))
                        count++;
                }
                if(i+1 < m){
                    if((j-1 >= 0) && ((board[i+1][j-1] == 1) ||
(board[i+1][j-1] == -2)))
                        count++;
                    if((board[i+1][j] == 1) || (board[i+1][j] == -2))
                        count++;
                    if((j+1 < n) && ((board[i+1][j+1] == 1) ||
(board[i+1][j+1] == -2)))
                        count++;
                }
                if((j-1 >= 0) && ((board[i][j-1] == 1) || (board[i][j-1] == -
2)))
                    count++;
                if((j+1 < n) && ((board[i][j+1] == 1) || (board[i][j+1] == -
2)))
                    count++;

                //Updating the value of the grid based on the live count.
                if(count == 3 && board[i][j] == 0)
                    board[i][j] = 2;
                else if(count < 2 && board[i][j] == 1)
                    board[i][j] = -2;
                else if(count > 3 && board[i][j] == 1)
                    board[i][j] = -2;
            }
        }

        //Loop to update the value by special identification number.
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                if(board[i][j] == -2)
                    board[i][j] = 0;
                else if(board[i][j] == 2)
                    board[i][j] = 1;
            }
        }
    }
```

17. Write a program that returns the next permutation of the number given in the array.

For example - the next permutation of arr = [1,2,3] is [1,3,2].

For solving this problem, we can find the peak element and then search for the element that has the next lowest element. Then we can sort the element. The element will be the next permutation.

```
public void next permutation(int[] nums) {
       int peakIndex = 0;
       //Finding the peak index
       for(int i = 1; i < nums.length; i++){
           if(nums[i] > nums[i-1])
               peakIndex = i;
       }
       //When the index is 0 the next permutation
       //will be the first. So return by sorting.
       if(peakIndex == 0){
           Arrays.sort(nums);
           return;
       }
       //Adjusting Pointers
       int prevIndex = peakIndex - 1;
       int index = prevIndex;
       int currIndex = peakIndex;

       //Finding the next greater element index
       while(currIndex < nums.length && nums[currIndex] > nums[index]){
           prevIndex = currIndex;
           currIndex++;
       }
       //swapping the value of that location
       int temp = nums[index];
       nums[index] = nums[prevIndex];
       nums[prevIndex] = temp;

       //Changing the values by sorting
       Arrays.sort(nums, index+1, nums.length);
   }
```

18. Given array nums containing n numbers of distinct values in the range [0, n], return the only value that is missing from the array.

We can follow the approach of mathematics by the formula of (n*n + n) / 2. This calculates the actual sum that needs to be in the array. And if we sum the array. Then the difference will be the answer.

```
//Method that returns the missing value
public int missingNumber(int[] nums) {
       int sumArray = 0;
       int n = nums.length;
       // Finding the sum of the array
       for(int i = 0; i<n; i++)
           sumArray += nums[i];
       // Calculating actual sum.
       int ActualSum = (n*n + n)/2;
       return ActualSum - sumArray;
```

```
    }
```

19. There is a loop in a linked list if there is some node where the next pointer can be followed to reach the same node again. The index pos is used to denote the node that the tail's next pointer is connected to. Note that pos is not passed as a parameter.

Given the linked list, return true if there is a cycle in the linked list. Otherwise, return false.

We can use a fast and slow pointer approach and will start traversing the linked list. The slow pointer will move 1 step forward and the fast pointer will move 2 steps forward. If there is a cycle in a linked list, then there will be a time when both the pointers will reach the same position. Then we will return true. Otherwise, if the pointer reaches the null then that means there was no loop in the linked list and then we will return false.

```
//Method that detects the loop in the linked list.
   public boolean hasCycle(ListNode head) {
       if(head == null || head.next == null)
           return false;

       //Managing slow and fast pointers.
       ListNode slow = head, fast = head;
       /*
       moving the pointers slow by 1 position
       and fast pointer by 2. If they meet that
       means there is a cycle. So return true.
       */
       while(fast != null && fast.next != null){
           fast = fast.next.next;
           slow = slow.next;
           if(slow == fast)
               return true;
       }
       //Otherwise return false if the linked list reaches the end
       return false; }
```

20. Given an array of distinct integer candidates and the desired integer target, produce a list of all the possible combinations of candidates where the sum of the chosen numbers is the target. You may return the combinations in any order.

You may choose the same number from the candidates array an unlimited number of times. Whether two combinations are unique depends on the frequency of at least one of the chosen numbers.

Example - Input: candidates = [2,3,6,7], target = 7

Output: [[2,2,3],[7]]

This problem seems that we can solve using the backtracking approach. So the solution will be.

```
class Solution {
   List<List<Integer>> ans;
   //Helper method that generates the result.
   private void findSum(int i, int target, List<Integer> consider, int[]
candidates){
       //If reaches the target then add one of the possible answers
       //to the answer array and terminate the recursive call

       if(target == 0){
           ans.add(new ArrayList<>(consider));
           return;
       }

       //If the index array is out of bound then terminate the call.
       else if(i == candidates.length)
           return;

       //If the target goes negative then also terminate the call.
       else if(target < 0)
           return;
       else{

           //Skipping the element and finding the sum with the next
           //Sub problem
           findSum(i+1, target, consider, candidates);

           //Adding the element to the list and trying the possibility
           //with the element and recursively calling for the next
           //sub-problem
           consider.add(candidates[i]);
           findSum(i, target-candidates[i], consider, candidates);

           //Backtracking. Removing the element from the list and
           //recursively calling for solving the next sub-problem.
           consider.remove(consider.size() - 1);
       }

   }
   public List<List<Integer>> combination(int[] candidates, int target) {
       ans = new ArrayList<>();

       //Calling helper function for solving the problem
       findSum(0, target, new ArrayList<>(), candidates);
       return ans;
   }
}
```

21. The array nums have an integer element at each position. At each position in the array, the element represents the length of the maximum jump you can from the position.

You are initially in the first position. Write a program that returns true if it reaches the last index from the first index. Otherwise, return false.

Example -

1. **Input: nums = [2,3,1,1,4]**
   Output: true
   In the above example. We can reach the last index either from the 0 indexes or 1 index.
2. **Input: nums = [3,2,1,0,4]**
   Output: false
   In the above example, we cannot reach the last index if any steps start from 0 so will return false.

This problem seems to be a Dynamic programming problem. So we can try all the possibilities by using the top-down approach and use memoization to optimize the recursive calls. So the solution is-

```
class Solution {
   int[] nums;
   Boolean[] dp;
   //Helper Method that returns the result
   private boolean solution(int i){
       //Base Cases
       if(i == nums.length-1)
           return true;
       if(i >= nums.length)
           return false;

       //Returning already computed result.
       //Overlapping Problems

       if(dp[i] != null)
           return dp[i];

       //Trying all the possibilities by going to
       //every step from the first index

       for(int j = 1; j <= nums[i]; j++){
           boolean choise = solution(i+j);
           //Memoization. Storing the computed result
           //So that it cannot be recomputed.

           if(choise) return dp[i] = true;
       }
       return dp[i] = false;
```

```
    }
    public boolean canJump(int[] nums) {
        this.nums = nums;
        dp = new Boolean[nums.length+1];
        //Calling helper method to set the answer.
        boolean ans = solution(0);
        return ans;
    }
}
```

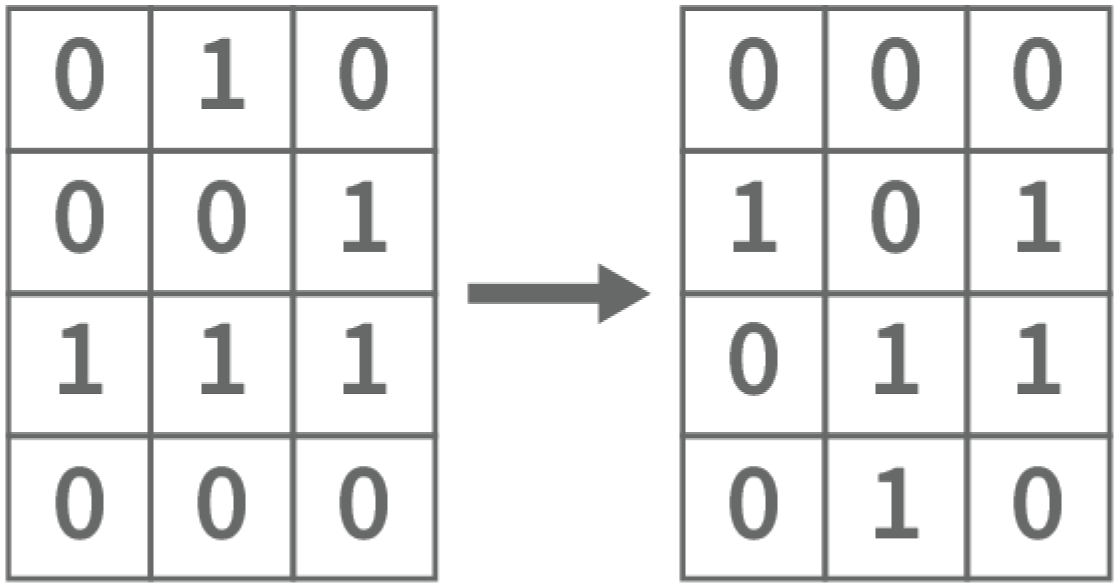
22. Write a program that returns true if the string contains the same frequency in for every character. Otherwise, return false.

We can solve this problem by counting the frequency in the hash and can check if all the value in the hash is the same or not.

```
public boolean areOccurrencesEqual(String s) {
        int max = 0;
        char[] hash = new char[26];
        //Counting the frequency and getting the max frequency
        for(int i = 0; i < s.length(); i++){
            hash[s.charAt(i)-'a']++;
            max = Math.max(max, hash[s.charAt(i)-'a']);
        }

        //Checking if the frequency matches with the max number
        for(int i = 0; i < 26; i++){
            if(hash[i] != 0 && hash[i] != max)
                return false;
        }
        return true;
    }
```

23. Write a program to reverse the singly linked list.



We can use a recursive approach to reverse the linked list. We can use head recursion and while returning time, we can adjust the pointers. The Solution is -

```
class Solution {
   private ListNode reverse(ListNode head){
       if(head == null || head.next == null){
           return head;
       }
       // Recursively calling function that returns the linked list for sub
problem
       ListNode x = reverse(head.next);
       //Adjusting the pointers
       head.next.next = head;
       head.next = null;
       return x;
   }
   public ListNode reverseList(ListNode head) {
       //Calling helper method that returns reversed liked list
       return reverse(head);
   }
}
```

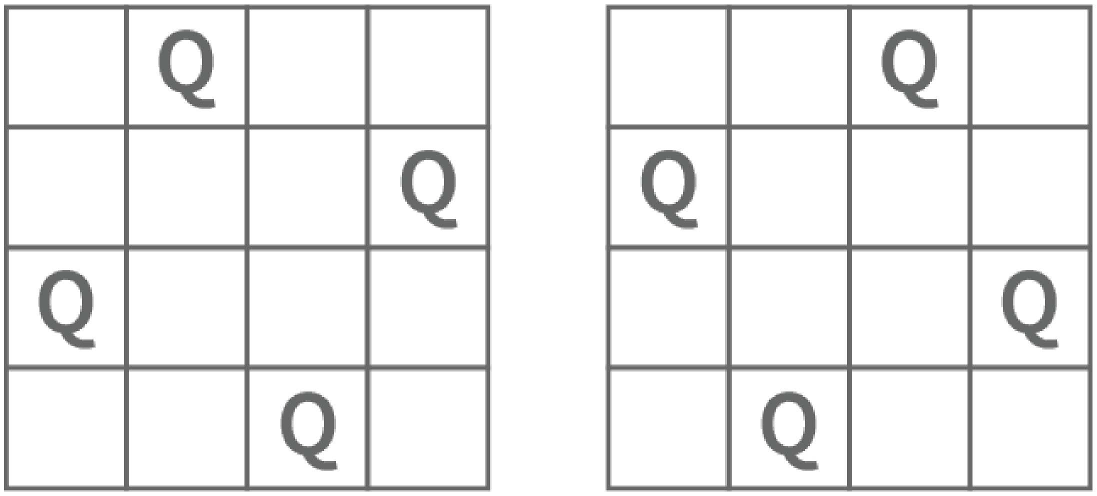
24. Write a program to print the nth Tribonacci number. A Tribonacci number is a number that has the value by adding the previous 3 values.

We can follow the approach of dynamic programming by storing the last 3 values and updating the values on iteration. The solution is -

```
class Solution {
   int trib(int n, int [] dp){
       //Base case for terminating the recursive call
       if(n == 0 || n == 1)
```

```
            return n;
        if(n == 2)
            return 1;
       //Finding the result not already computed
                    if(dp[n] == 0){
            dp[n] =  trib(n-1, dp) + trib(n-2, dp) + trib(n-3, dp);
        }
        //Returning already computed result or the result that
        //latest computed
        return dp[n];
    }
    public int tribonacci(int n) {
        int []dp = new int[n+1];
        //Calling helper method that computes and returns the result.
        return trib(n, dp);
    }
}
```

25. Write a program to solve the n-queen problem. You will be given an N x N board. And the task is to place the queen in a position such that it cannot overlap the queen diagonally, vertically, or diagonally.



For input 4, there can be 2 possible solutions, So you need to return that.

We can solve this problem with the help of backtracking. We can try every possibility and we can check if the queen does not get overlapped. So the solution will be -

```
class Solution {
    //HashMap that keeps track of the attacking position of the queen
    HashMap<Integer, Boolean> posDiag;
    HashMap<Integer, Boolean> negDiag;
    HashMap<Integer, Boolean> vertical;
    //List that stores the possible answers
    List<List<String>> ans;
    int N;
```

```
   // Helper method that generates the answer
   private void solution(int i, List<String> res){
       //If it reaches the end then that will be a possible solution.
       //So we will add this to the list.
       if(i == N){
           ans.add(new ArrayList<String>(res));
           return;
       }
       // Checking if every position is not attacked by checking from
       // hashmap. And if not attacking then the recursive calling the next
solution.
       for(int j = 0; j < N; j++){
           int posIndex = i+j;
           int negIndex = i-j;
           if(posDiag.containsKey(posIndex) ||
              negDiag.containsKey(negIndex) ||
              verticle.containsKey(j)) continue;

           posDiag.put(posIndex, true);
           negDiag.put(negIndex, true);
           //Adding the possible answer to the answer list
           verticle.put(j, true);
           String old = res.get(i);
           StringBuffer sb = new StringBuffer(old);
           sb.replace(j, j+1, "Q");
           res.set(i, sb.toString());

           // Recursively calling the method to generate an answer.
           solution(i+1, res);

           //Removing queen and trying the next possibility.
           // Backtracking.
           posDiag.remove(posIndex);
           negDiag.remove(negIndex);
           verticle.remove(j);
           res.set(i, old);
       }
   }
   public List<List<String>> solveNQueens(int n) {
       ans = new ArrayList<>();
       if(n == 2 || n == 3)
           return ans;
       posDiag = new HashMap<>();
       negDiag = new HashMap<>();
       verticle = new HashMap<>();
       N = n;
       StringBuffer sb = new StringBuffer();
       List<String> res = new ArrayList<>();
       for(int i = 0; i < n; i++) sb.append(".");
       for(int i = 0; i < n; i++) res.add(new String(sb.toString()));

       //Calling helper method to generate the solution.
       solution(0, res);
       return ans;
   }
}
```

Frequently Asked Questions

1. Is the Nagarro interview hard?

Online coding test in Nagarro is considered hard level. Because they ask 5 questions from the graph or trees in a limited time. But if you have already practised enough, then it’s not too difficult to pass all the test cases. And also if you cannot solve all the 5 questions but you managed to solve 4 out of 5 then you get a high chance to clear for the next round. An in-person technical interview is also of a medium level. You just need to practice and give answers confidently.

2. What is your biggest achievement?

This is the common question that will be asked by your interviewer during your interview. The best approach to answer this question is to consider picking something that is as recent as possible and relevant to this job or your career. Even if they don't specifically ask for one, pick a professional achievement.

Consider an example answer for freshers- My most significant professional accomplishment was finishing my engineering degree in four years with an 8.4 SGPA. I had no financial backing from my family and had to work full-time so that I might study engineering. This taught me to be organized, maintain good habits, and focus on my goals. I am pleased with my accomplishment, and I think that my lessons will help me in my future career.

3. How to crack the Nagarro aptitude test?

There is no specific trick that can be used to crack the Nagarro Aptitude Test. The only way is to practice questions. You can refer to the asked questions from the above section or previous years.

4. What is the salary for a fresher in Nagarro?

Nagarro offers a salary package in India-

Profile	Salary (in CTC)
Associate Software Developer	4.5 LPA
Software Engineer	6 - 6.5 LPA
Senior Software Engineer	12-15 LPA

5. How long is the Interview Process at Nagarro?

The complete process starting from the online test to the job offer takes **almost 20-25 days**. After each round, the company takes 2-3 business days to release the results. And after the HR interview, generally, it depends on the hiring team how much time they take to release the offer letter. But generally, it will be reached within 1 week.

6. What is the Eligibility Criteria at Nagarro?

- **Course Eligibility:** B.Tech (Computer Science & Engineering, Information Technology, and Electronics and Communication Engineering/ MCA.
- **Percentage Criteria:** Overall greater than 60%.
- **Backlog:** No Active Backlog.
- **Year Gap:** Not more than 2 years.