**Data Structures:**

**Write the benefits of using an algorithm on O(n) vs O(log n) time complexity.**

When comparing algorithms with **O(n)** (linear time complexity) and **O(log n)** (logarithmic time complexity), the advantages of each depend on how they're used. Here’s a simpler explanation of the benefits:

**Benefits of O(log n):**

1. **Faster for Large Inputs:**
   * Algorithms with O(log n) grow very slowly as the input size increases. Even with large data, they perform relatively few operations.
   * Example: In a sorted list, binary search finds an item quickly by repeatedly cutting the list in half.
2. **Handles Big Data Better:**
   * These algorithms work well with larger datasets. Doubling the size of the data only slightly increases the number of steps.
3. **Great for High-Speed Needs:**
   * Perfect for applications like databases or real-time systems where efficiency and speed are important.
4. **Useful in Problem-Splitting Approaches:**
   * These algorithms often split the task into smaller parts (like searching in a tree), making them ideal for structured data.

**Benefits of O(n):**

1. **Easy to Understand and Write:**
   * Linear algorithms are usually straightforward and simpler to implement.
   * Example: Going through a list to find the largest number is a basic O(n) task.
2. **Best for Processing All Data:**
   * If you need to work with every item in a dataset, O(n) ensures no element is skipped.
   * Example: Adding up all the numbers in a list.
3. **Lower Complexity Overhead:**
   * They don’t rely on extra steps or structures like recursion, making them efficient in simpler cases.
4. **Good for Small Datasets:**
   * For small inputs, the speed difference between O(n) and O(log n) is minimal. Using O(n) is often easier and just as effective.

**When to Use Each:**

* Choose **O(log n)** when:
  + You can break the problem into smaller parts (like searching in sorted data or trees).
  + You need speed and efficiency, especially with large datasets.
* Choose **O(n)** when:
  + Every item in the dataset needs to be processed.
  + The problem is straightforward, and simplicity matters.
  + The dataset is small, where the time complexity difference doesn’t make a big impact.

**Benefits of using Stacks and Queues:**

**Benefits of Using Stacks**

1. **Easy to Manage Data:**
   * Stacks follow the Last In, First Out (LIFO) rule, meaning the last item added is the first one to be processed. This makes them perfect for managing tasks where the newest item is handled first**.**
2. **Reversing Things:**
   * Stacks are useful for reversing items, like flipping a string or list.
3. **Managing Function Calls:**
   * Stacks are used in programming to handle function calls, recursion, and tasks that require backtracking. For example, the "call stack" in programming keeps track of which function is currently running.
4. **Undo Actions:**
   * Stacks are great for adding undo features in applications like text editors, where recent changes can be rolled back.
5. **Math and Syntax:**
   * Stacks help evaluate mathematical expressions (like postfix or prefix notation) and parse code in compilers.

**Benefits of Using Queues**

1. **Organized Processing:**
   * Queues follow the First In, First Out (FIFO) rule, meaning the first item added is the first one to be processed. This is ideal for tasks that need to happen in order.
2. **Scheduling and Task Handling:**
   * Queues are used in operating systems to manage tasks like scheduling processes and handling requests.
3. **Managing Streams:**
   * Queues handle streaming data in real-time, like buffering videos or sending messages between systems.
4. **Breadth-First Search (BFS):**
   * Queues are used in BFS algorithms for exploring graphs or trees level by level.
5. **Handling Multiple Requests:**
   * They are essential for web servers and APIs, ensuring requests are handled in the order they arrive.

**When to Use Stacks**

* When you need to process the most recent item first.
* For tasks like recursion, backtracking, or reversing data.
* In applications with undo or rollback features.

**When to Use Queues**

* When tasks need to be handled in the order they arrive.
* For scheduling tasks, managing data streams, or handling requests one by one.
* In algorithms like BFS or for real-time data processing.

**Leet Code:179**

class Solution {

    public String largestNumber(int[] nums) {

        List<String> list=new ArrayList<>();

        for(int i: nums){

            list.add(String.valueOf(i));

        }

        Collections.sort(list, (a,b)->compare(a,b));

        String ans=String.join("", list);

        if(allZeros(ans)) return "0";

        return ans;

    }

    private boolean allZeros(String s){

        for(char c: s.toCharArray()){

            if(c!='0') return false;

        }

        return true;

    }

    private int compare(String a, String b){

        int i=0, j=0;

        while(i<a.length() || j<b.length()){

            if(i==a.length()) i=0;

            if(j==b.length()) j=0;

            if(a.charAt(i)>b.charAt(j)) return -1;

            if(a.charAt(i)<b.charAt(j)) return 1;

            i++;

            j++;

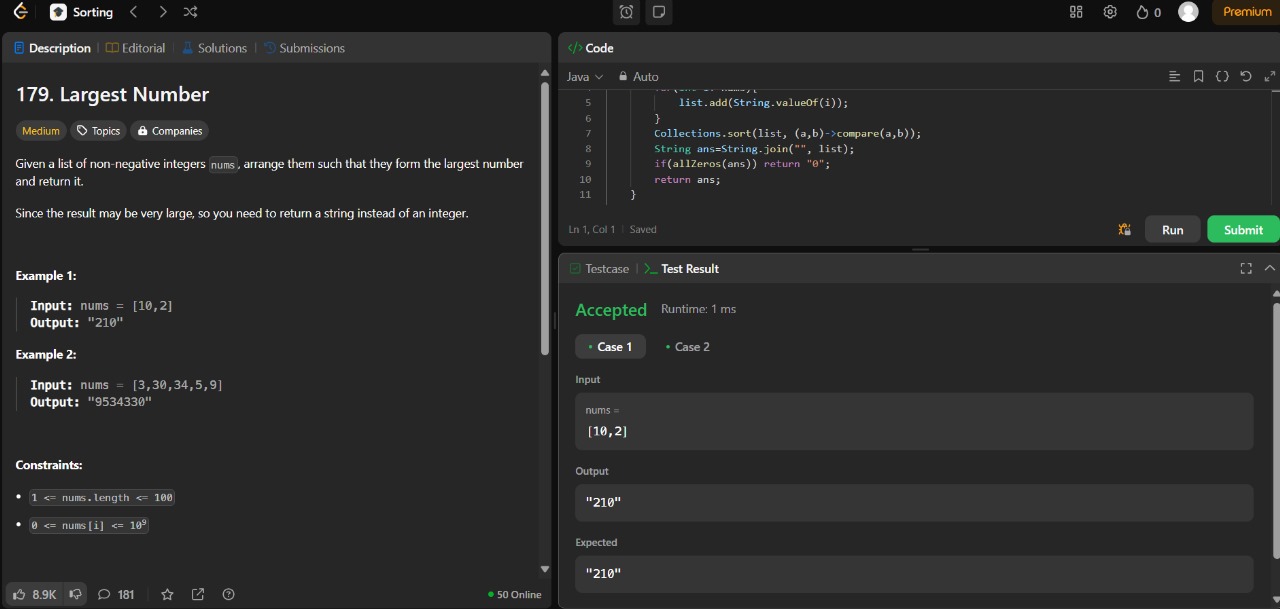
        }

        return 0;

    }

}

**Output:**



**Leet code 21:**

class Solution {

public ListNode mergeTwoLists(ListNode l1, ListNode l2) {

ListNode dummy = new ListNode(0);

ListNode curr = dummy;

while(l1!=null && l2!=null){

//Merge List

if(l1.val <= l2.val) {

curr.next = l1;

l1=l1.next;

}

else{

curr.next = l2;

l2=l2.next;

}

curr= curr.next;

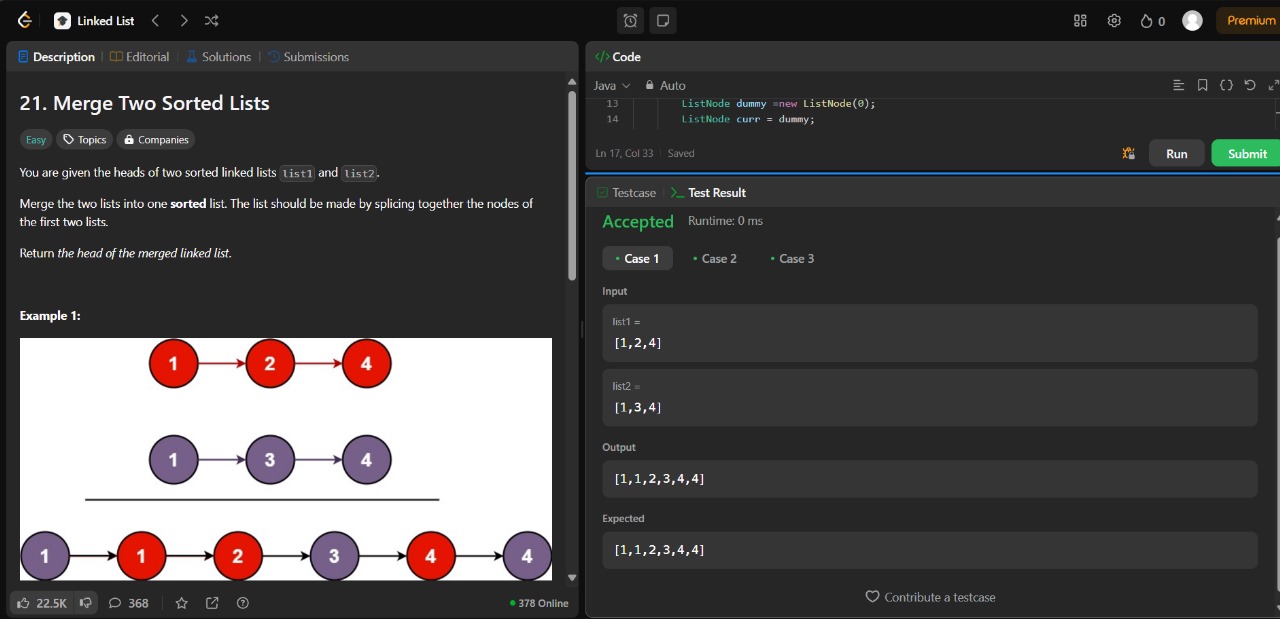
}

curr.next= l1!=null ? l1 : l2;

return dummy.next;

}}

**Output:**



**Content Writing:**

**Write an email requesting mentorship from a founder in a field of your interest.**

Subject: Request for Mentorship in [XXX]

Dear [XYZ],

I hope this email finds you well. My name is [YYY], and I am deeply inspired by your work at [Fullstack project] in the field of [Software]. Your journey and achievements have motivated me to reach out and learn from your expertise.

I am a recent graduate specializing in software development and have experience on Fullstack Developer for 6months and now, I aspire to grow in [Software] and believe that your guidance could provide invaluable insights and direction.

I understand that your time is incredibly valuable, and I truly appreciate the effort it takes to mentor someone. If you're open to it, I would be grateful for the opportunity to connect—whether through a brief call, or advice via email.

Thank you for considering this request. Please let me know if there's a time and format that works best for you, or if there are any prerequisites I should prepare.

Looking forward to the possibility of learning from you.

Thanks $ Regards,

[YYY]

[1234567843]

**Interview Process Report: WAD**

**Date of Interview:** [5/12/24]  
**Position Applied For:** [Fullstack Developer]  
**Interview Mode:** [Online]  
**Interviewer(s):** [Ruchika Mam]

**Stages of the Interview Process**

**1. Application Submission**

* **Details:** Submitted my application through [platform/medium, e.g., company website, job portal].
* **Documents Submitted:** Resume, cover letter, and portfolio links (if applicable).

**2. Initial Screening**

* **Type:** Telephonic/Online Questionnaire.
* **Duration:** [15 minutes].
* **Key Areas:**
  + Discussion about my background, skills, and why I’m interested in WAD.
  + Brief overview of my previous projects and their relevance to the position.

**3. Technical Round**

* **Mode:** [Online].
* **Duration:** [30 minutes].
* **Key Focus Areas:**
  + **Technical Skills:** Practical questions related to [specific technologies/tools required for the role].
  + **Problem-Solving:** Hands-on coding exercises or case studies.
  + **Project Discussion:** In-depth review of my past projects, including challenges faced and solutions implemented.

**4. HR/Behavioural Round**

* **Mode:** [Online/Offline].
* **Duration:** [30 minutes].
* **Topics Discussed:**
  + Alignment of my goals with WAD's mission and values.
  + Strengths, weaknesses, and scenarios showcasing teamwork and leadership.
  + Salary expectations and availability.

**Key Takeaways**

* The interview process was thorough, with a strong focus on [specific aspects, e.g., problem-solving, adaptability, teamwork].
* The interviewers were professional, supportive, and open to questions about the role and the company.
* I gained a deeper understanding of WAD’s vision and how I can contribute to its success.

**Outcome and Feedback**

* **Result:** [Selected/Rejected/Waiting for Results].
* **Feedback Received:** [If applicable, write the feedback you received].

**Reflection**

The interview process with WAD was an excellent learning opportunity. It reinforced my understanding of [specific skills or concepts], and I am eager to apply these insights in future opportunities.

Warm regards,  
[YYY]

**Color Palette:**

1. Navy blue is a classic, sophisticated color that pairs beautifully with a variety of shades.

2. Sky blue and white make for a refreshing and clean combination.