Assume you are expert in Java and Problem solver then provide roadmap to be expert in list methods and able to solve any Problems

Absolutely, Routh Kiran! Since you're diving deep into Java and aiming to **become an expert in `List` methods** and **solve any problem using lists**, here's a **step-by-step roadmap** tailored to your goal.

🎯 Goal:

> **Become an expert in Java List methods and solve any list-related problems confidently.**

✅ Phase 1: Foundations of Java Collections

📌 Step 1: Understand List Basics

- What is a `List` in Java?
- Difference between `List`, `Set`, and `Map`
- List is an **interface**, implemented by:
    - `ArrayList`
    - `LinkedList`
    - `Vector`
    - `Stack`

✅ Practice:

java
CopyEdit

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

✅ Phase 2: Master All Important `List` Methods

📘 Step 2: Learn These Common Methods with Examples

Method	Description	Example
`add(E e)`	Add element	`list.add("Java")`
`add(int index, E element)`	Insert at index	`list.add(1, "Expert")`
`get(int index)`	Get element	`list.get(0)`
`set(int index, E element)`	Replace element	`list.set(0, "Python")`
`remove(int index)`	Remove by index	`list.remove(2)`
`remove(Object o)`	Remove first occurrence	`list.remove("Java")`
`contains(Object o)`	Check existence	`list.contains("Java")`
`size()`	Total elements	`list.size()`
`isEmpty()`	Is list empty	`list.isEmpty()`
`clear()`	Remove all elements	`list.clear()`
`indexOf(Object o)`	First index of element	`list.indexOf("Java")`
`lastIndexOf(Object o)`	Last index of element	`list.lastIndexOf("Java")`
`subList(from, to)`	Get sub list	`list.subList(1, 3)`

`toArray()`	Convert to array	`list.toArray()`

✅ Practice:

Create a notebook/project and write small code examples for each method. Test different data types like `Integer`, `String`, `Custom Objects`.

✅ Phase 3: Learn List Implementations in Detail

📗 Step 3: Know When to Use Which List

Type	When to Use
`ArrayList`	Fast access, frequent reads
`LinkedList`	Frequent add/remove at ends
`Vector`	Thread-safe (legacy)
`Stack`	LIFO operations

Explore:

- Time Complexity of operations (`add`, `get`, `remove`)
- Memory behavior

✅ Phase 4: Practice List Problems Daily

💪 Step 4: 100+ Problems (LeetCode / HackerRank / Masai Platform)

🎯 Start With:

- Insert & Delete

- Find min/max
- Reverse list
- Merge two lists
- Frequency of elements

🎯 **Move To:**

- Remove duplicates
- Rotate list
- Sublist search
- Palindrome list
- Find missing/duplicate numbers

🎯 **Advanced:**

- Custom sorting using `Comparator`
- `List<List<Integer>>` problems (Matrix style)
- Streams & Lambda expressions on Lists

✅ Phase 5: Master List Utilities from Java Libraries

🧰 Step 5: Use `Collections` & `Stream` API

- `Collections.sort(list)`
- `Collections.reverse(list)`
- `Collections.frequency(list, item)`

Streams:

java
CopyEdit
```
list.stream().filter(x -> x > 10).collect(Collectors.toList());
```

-

✅ Phase 6: Build Projects Using Lists

👨‍💻 Mini Projects

- Student Management System
- To-Do List
- Custom Stack using List
- Shopping Cart System

These will give real-world context.

✅ Phase 7: Mock Interviews + Real Challenges

- Practice DSA (Data Structures & Algorithms)
- Participate in contests
- Solve problems with time/space optimization

Bonus Tips:

🔹 Use **debugger** to trace list changes
🔹 Create **mind maps** of methods
🔹 Discuss problems with peers or explain to a rubber duck (Rubber duck debugging!)

- Learn how `List` works under the hood (Array resizing in `ArrayList`, node linking in `LinkedList`)