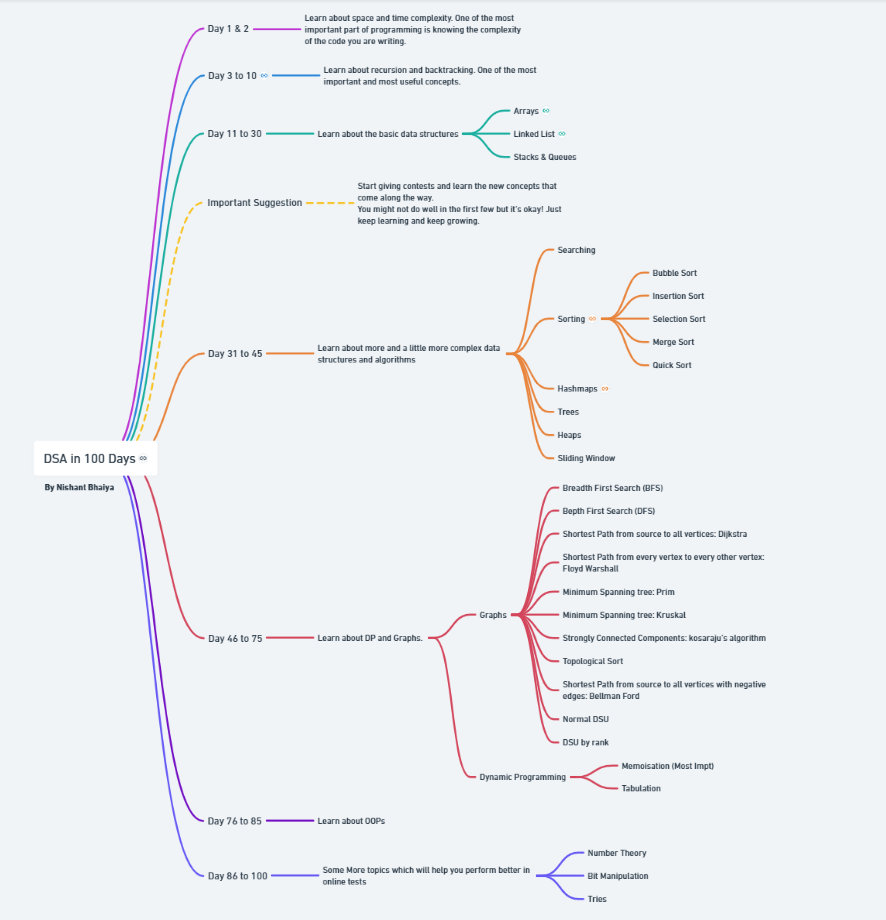
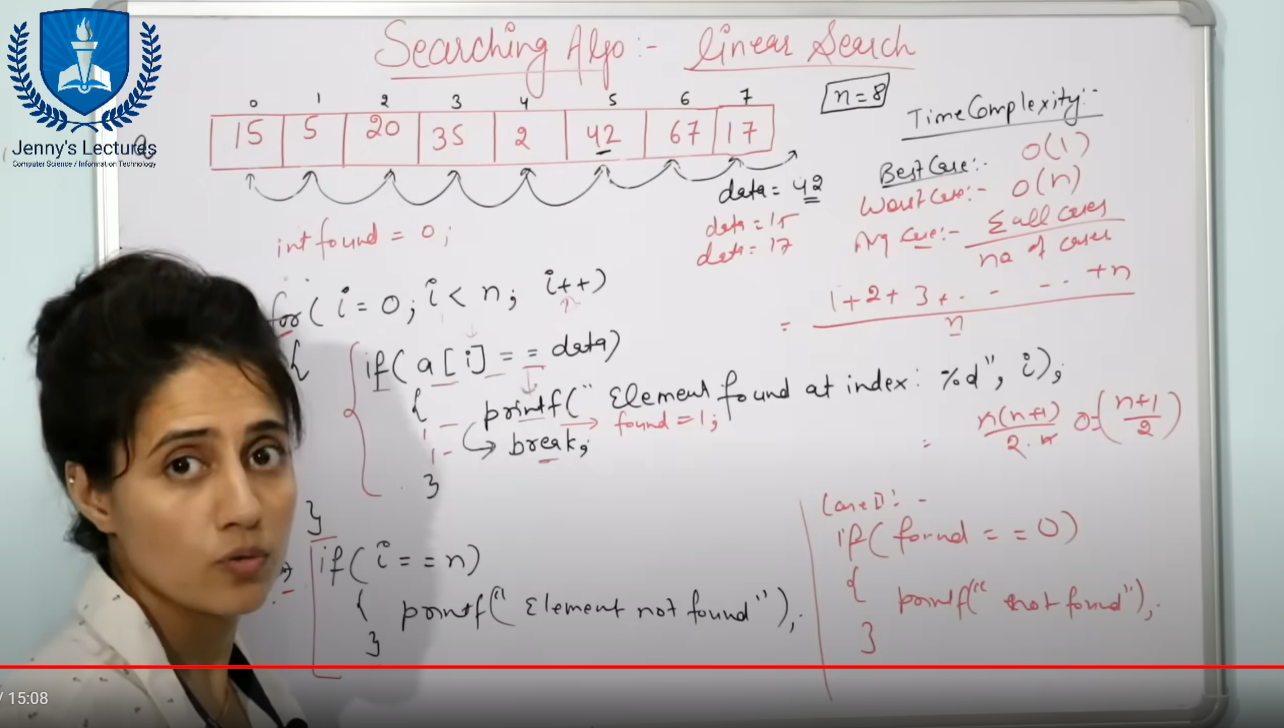
**DSA AND Analysis of Algorithm**

**Roadmap:-  
   
Module1:** **Introduction to Algorithms  
  
  
Module1:** **Analysis of Basic Algorithms** **Code:-***/\*\**

*\* The above Java class implements a linear search algorithm to find a given element in an array.*

*\*/*  
import *java.util.Scanner*;

*public* *class* Linear\_Searching\_Algo {

*public* *static* void main(String args[])

    {

        Scanner sc= new Scanner(System.in);

        System.out.print("Enter the length of the Array:");

        int n=sc.nextInt();

        int Array[]=new int[n];

        int i;

        for(i=0;i<n;i++)

        {

            System.out.print("Enter the element of the array:");

            Scanner sc1=new Scanner(System.in);

            Array[i]=sc1.nextInt();

        }

        System.out.println("The Elements of the Array are as follows:");

        for(i=0;i<n;i++)

        {

            System.out.println(Array[i]);

        }

        Scanner sc2=new Scanner(System.in);

        System.out.println("Enter the number you want to find in the array:");

        int element=sc2.nextInt();

        int found=0;

        for(i=0;i<n;i++)

        {

            if(Array[i] == element)

            {

                System.out.println("Element found at index:" + i);

                found=1;

            }

        }

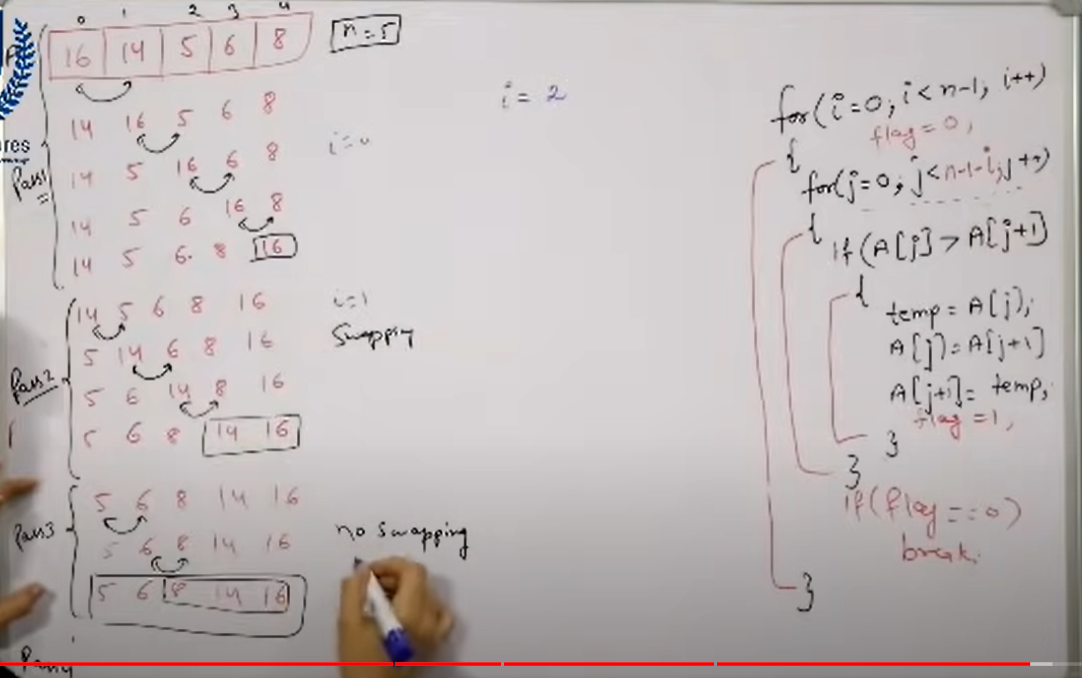
         if(found != 1)

            {

                System.out.println("The element was not found in the array");

            }

    }

}   
  
  
  
  
**we use the flag variable so that if the array is already sorted we don’t waste iterations in trying to sort the already sorted array.  
  
Worst Time complex city:- o(n2).  
Best Time complex city:- o(n).**  
  
**Code:-**  
*/\*\**

*\* The Bubble\_Sor\_Algo class implements the bubble sort algorithm to sort an array of integers in*

*\* ascending and descending order.*

*\*/*

import *java.util.Scanner*;

*public* *class* Bubble\_Sor\_Algo {

*public* *static* void main(String args[])

    {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the Length of the array:");

        int n=sc.nextInt();

        int Array[]= new int [n];

        int i;

        for(i=0;i<n;i++)

        {

            Scanner sc1 = new Scanner(System.in);

            System.out.print("Enter the element of the array:");

            Array[i] =sc1.nextInt();

        }

        System.out.println("The Elements of the Array are as follows");

        for(i=0;i<n;i++)

        {

            System.out.println(Array[i]);

        }

        int j;

        int temp;

        int flag;

        for(i=0;i<n-1;i++)

        {

            flag=0;

            for(j=0;j<n-i-1;j++)

            {

                if(Array[j]>Array[j+1])

                {

                   temp=Array[j];

                   Array[j]=Array[j+1];

                   Array[j+1]=temp;

                   flag=1;

                }

                if(flag ==0)

                {

                    break;

                }

            }

        }

        System.out.println("The Elements of the Sorted Array in Accending order is are as follows");

        for(i=0;i<n;i++)

        {

            System.out.println(Array[i]);

        }

        for(i=0;i<n-1;i++)

        {

            for(j=0;j<n-i-1;j++)

            {

                if(Array[j]<Array[j+1])

                {

                   temp=Array[j];

                   Array[j]=Array[j+1];

                   Array[j+1]=temp;

                }

            }

        }

        System.out.println("The Elements of the Sorted Array in Accending order is are as follows");

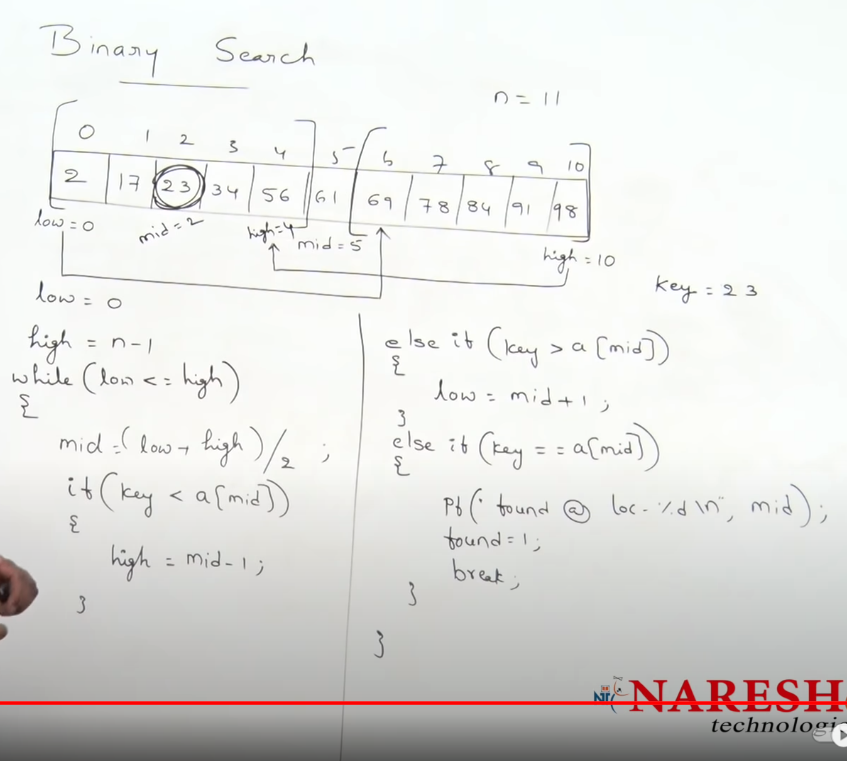
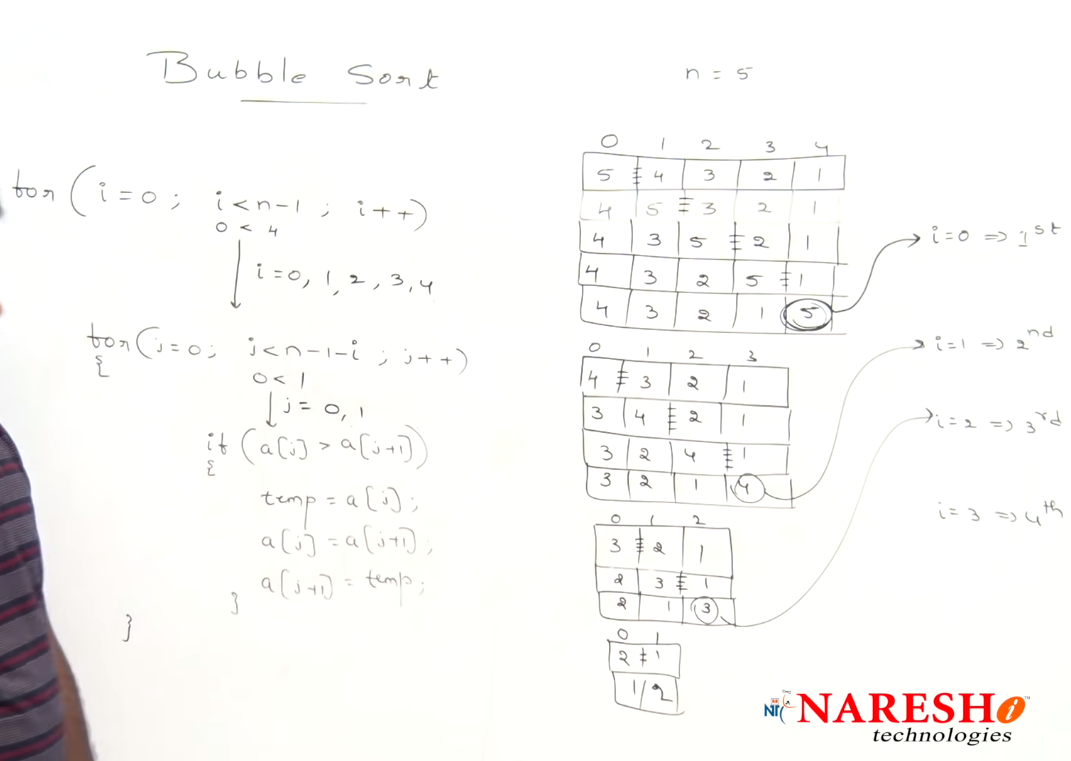
        for(i=0;i<n;i++)

        {

            System.out.println(Array[i]);

        }

    }

}  
  
  
  
  
**Code:-**  
*/\*\**

*\* The Binary\_Search\_Algo class implements a binary search algorithm to search for an element in a*

*\* sorted array.*

*\*/*  
import *java.util.Scanner*;

*public* *class* Binary\_Search\_Algo {

*public* *static* void main(String args[])

    {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the Length of the array:");

        int n=sc.nextInt();

        int Array[]= new int [n];

        int i;

        for(i=0;i<n;i++)

        {

            Scanner sc1 = new Scanner(System.in);

            System.out.print("Enter the element of the array:");

            Array[i] =sc1.nextInt();

        }

        System.out.println("The Elements of the Array are as follows");

        for(i=0;i<n;i++)

        {

            System.out.println(Array[i]);

        }

        int j;

        int temp;

        for(i=0;i<n-1;i++)

        {

            for(j=0;j<n-i-1;j++)

            {

                if(Array[j]>Array[j+1])

                {

                   temp=Array[j];

                   Array[j]=Array[j+1];

                   Array[j+1]=temp;

                }

            }

        }

        System.out.println("The Elements of the Sorted Array in Accending order is are as follows");

        for(i=0;i<n;i++)

        {

            System.out.println(Array[i]);

        }

        Scanner sc2 = new Scanner(System.in);

        System.out.print("Enter the element you want to find:");

        int element=sc2.nextInt();

        int found=0;

        int low=0;

        int high=n-1;

        while(low<=high)

        {

            int mid;

            mid=(high+low)/2;

            if(element < Array[mid])

            {

                high=mid-1;

            }

            else if(element > Array[mid])

            {

                low=mid+1;

            }

            else if(element == Array[mid])

            {

                System.out.println("The element is found at index:" + mid);

                found=1;

                break;

            }

        }

        if(found == 0)

            {

                System.out.println("The element is not present in the array");  
 for(i=0;i<n;i++)

                {

                    if(element<Array[i])

                    {

                        System.out.println("The element if it was present would be at index:" + i);

                        break;

                    }

                }

                if(i>=n)

                {

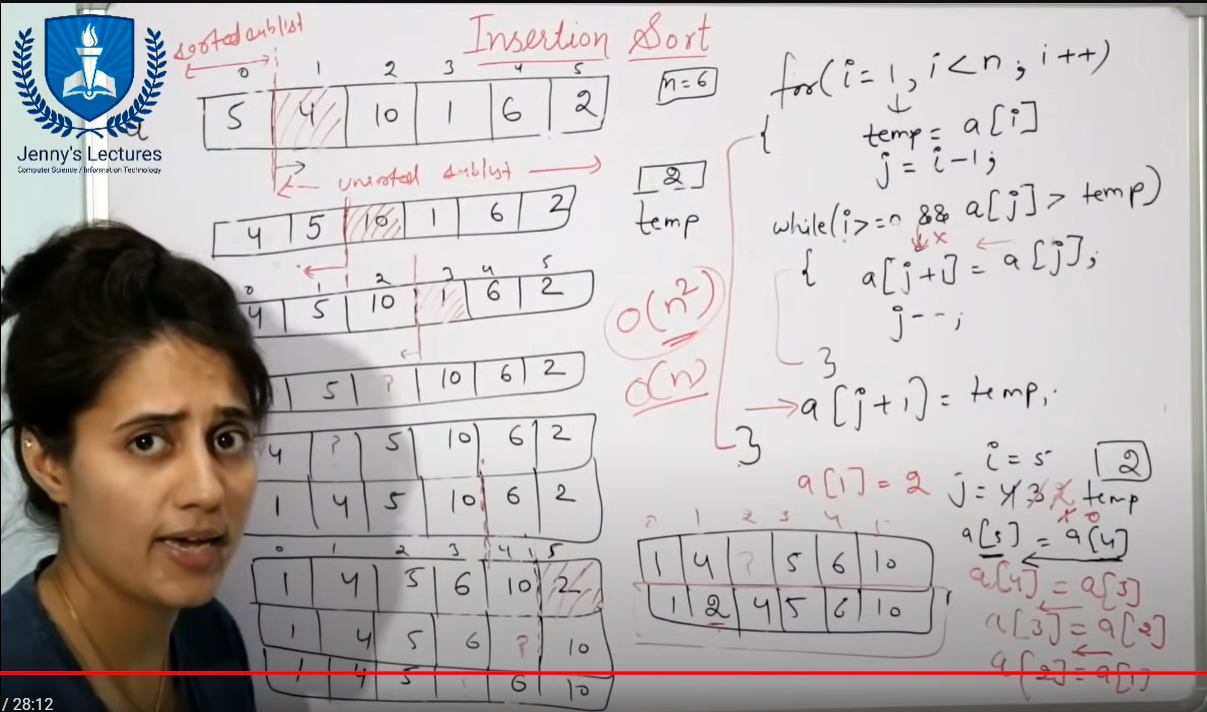
                    System.out.println("The element if it was present would be at index:" + n);

                }

            }

    }

}



Worst Time complex city:- o(n2).  
Best Time complex city:- o(n).  
  
  
**Code:-***/\*\**

*\* The Insertion\_Sort\_Algo class takes user input for an array, sorts it in ascending and descending*

*\* order using the insertion sort algorithm, and then prints the sorted arrays.*

*\*/*

import *java.util.Scanner*;

*public* *class* Insertion\_Sort\_Algo {

*public* *static* void main(String args[])

    {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the Length of the array:");

        int n=sc.nextInt();

        int Array[]= new int [n];

        int i;

        for(i=0;i<n;i++)

        {

            Scanner sc1 = new Scanner(System.in);

            System.out.print("Enter the element of the array:");

            Array[i] =sc1.nextInt();

        }

        System.out.println("The Elements of the Array are as follows");

        for(i=0;i<n;i++)

        {

            System.out.println(Array[i]);

        }

        int j;

        int temp;

        for(i=1;i<n;i++)

        {

            temp=Array[i];

            j=i-1;

            while(j>=0 && Array[j]>temp)

            {

                Array[j+1]=Array[j];

                j--;

            }

            Array[j+1]=temp;

        }

        System.out.println("The Elements of the Sorted Array in Accending order is are as follows");

        for(i=0;i<n;i++)

        {

            System.out.println(Array[i]);

        }

        for(i=1;i<n;i++)

        {

            temp=Array[i];

            j=i-1;

            while(j>=0 && Array[j]<temp)

            {

                Array[j+1]=Array[j];

                j--;

            }

            Array[j+1]=temp;

        }

        System.out.println("The Elements of the Sorted Array in Decending  order is are as follows");

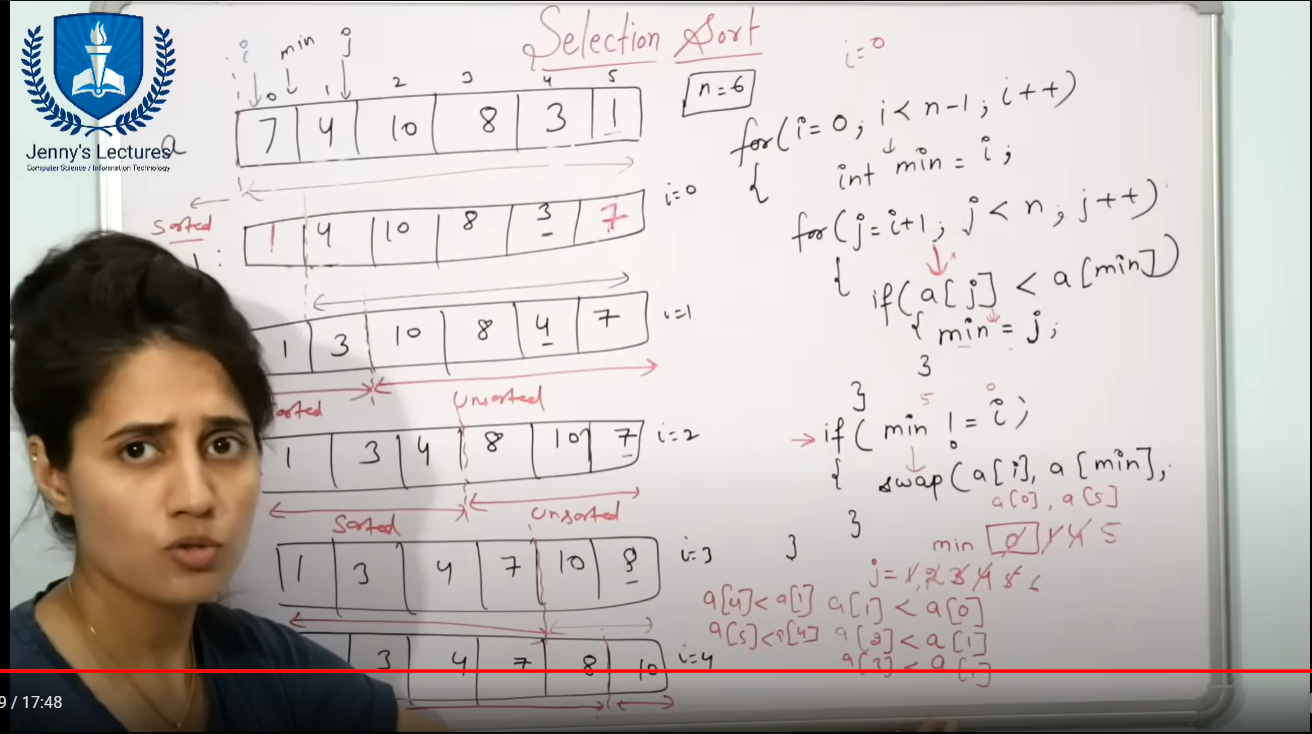
        for(i=0;i<n;i++)

        {

            System.out.println(Array[i]);

        }

    }

}  
  
  
  
Worst Time complex city:- o(n2).  
Best Time complex city:- o(n2).  
  
**Code:-**

*/\*\**

*\* The Selection\_Sort\_Algo class implements the selection sort algorithm to sort an array of integers*

*\* in ascending and descending order.*

*\*/*

import *java.util.Scanner*;

*public* *class* Selection\_Sort\_Algo {

*public* *static* void main(String args[])

    {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the Length of the array:");

        int n=sc.nextInt();

        int Array[]= new int [n];

        int i;

        for(i=0;i<n;i++)

        {

            Scanner sc1 = new Scanner(System.in);

            System.out.print("Enter the element of the array:");

            Array[i] =sc1.nextInt();

        }

        System.out.println("The Elements of the Array are as follows");

        for(i=0;i<n;i++)

        {

            System.out.println(Array[i]);

        }

        System.out.println("Following is the Sorted Array in Accending Order:");

        Selection\_Sort\_Asc(Array , n);

        System.out.println("Following is the Sorted Array in Decending Order:");

        Selection\_sort\_Dec(Array , n);

    }

*public* *static* void Selection\_Sort\_Asc(int [] Array , int n)

    {

        int i;

        int j;

        int min;

        int temp;

        for(i=0;i<n-1;i++)

        {

            min=i;

            for(j=i+1;j<n;j++)

            {

                if(Array[j]<Array[min])

                {

                    min=j;

                }

            }

            if(i != min)

            {

                temp=Array[i];

                Array[i]=Array[min];

                Array[min]=temp;

            }

        }

        for(i=0;i<n;i++)

        {

            System.out.println(Array[i]);

        }

    }

*public* *static* void Selection\_sort\_Dec(int [] Array , int n)

    {

        int i;

        int j;

        int max;

        int temp;

        for(i=0;i<n-1;i++)

        {

            max=i;

            for(j=i+1;j<n;j++)

            {

                if(Array[j]>Array[max])

                {

                    max=j;

                }

                if(i != max)

                {

                    temp=Array[i];

                    Array[i]=Array[max];

                    Array[max]=temp;

                }

            }

        }

        for(i=0;i<n;i++)

        {

            System.out.println(Array[i]);

        }

    }

}

Do form Apna college and Jenny CS\IT  
  
Array and Strings

Recursion Searching Sorting

cycle sort prefix sum sliding window

object oriented stacks queues linked lists

binary tree binary search trees heaps priority queues maps sets

backtracting dynamic programing graph avl trees segmentary  
  
  
DEVOPS:- [DevDocs API Documentation](https://devdocs.io/)  
To be considered excellent in Java for the Software Student Training in Engineering Program (STEP) Internship, you should have a strong understanding of the following:

1. **Core Java Concepts:**
   * Variables, data types, and operators.
   * Control flow statements (if, else, switch, loops).
   * Exception handling.
   * File I/O operations.
   * Threads and concurrency.s
2. **Object-Oriented Programming (OOP):**
   * Classes and objects.
   * Inheritance, polymorphism, encapsulation, and abstraction.
   * Interfaces and abstract classes.
   * Constructors and destructors.
3. **Java Libraries and Frameworks:**
   * Familiarity with commonly used Java libraries.
   * Understanding of Java Collections Framework (List, Set, Map, etc.).
   * Knowledge of utility classes (e.g., Collections, Arrays, etc.).
4. **Java-related Technologies:**
   * JDBC (Java Database Connectivity) for database interactions.
   * Servlets and JSP (JavaServer Pages) for web development (if mentioned in the internship requirements).
   * Basics of JavaFX or Swing for GUI development (if relevant).
5. **Development Tools:**
   * Proficiency in using development tools such as Eclipse, IntelliJ IDEA, or similar.
   * Understanding of version control systems (e.g., Git).
6. **Testing:**
   * Knowledge of testing frameworks (e.g., JUnit) for unit testing.
   * Familiarity with testing methodologies and practices.
7. **Best Practices and Design Patterns:**
   * Clean code principles.
   * Familiarity with design patterns (e.g., Singleton, Factory, Observer).
8. **Java-related Technologies and Tools:**
   * Maven or Gradle for project build and dependency management.
   * Logging frameworks (e.g., Log4j).
   * Application servers (e.g., Apache Tomcat).
9. **Concurrency and Multithreading:**
   * Understanding of Java concurrency mechanisms.
   * Experience with multithreaded programming.
10. **Web Development (if applicable):**
    * Basics of HTML, CSS, and JavaScript (if the internship involves web development).
11. **Problem-Solving and Algorithmic Skills:**
    * Ability to solve problems algorithmically using Java.

Remember that the specific requirements may vary depending on the company and the nature of the internship. It's always a good idea to review the company's technologies stack and any specific tools mentioned in the internship listing to tailor your preparation accordingly.