

DSA Study Notes Day 1

Created by Munawar Johar

How to Study This Series?

- **Understand the Concepts:** Before diving into code, ensure you understand the underlying concepts.
- **Practice Regularly:** Regular practice is essential for mastering DSA.
- **Follow the Flow:** Start with simple problems and gradually move to more complex ones.
- **Analyze Your Code:** Always consider the time and space complexity of your solutions.
- **Solve Multiple Problems:** Approach problems from different angles and solve them using various methods.

How to Solve Problems?

1. **Understand the Problem:** Make sure you clearly understand the problem before attempting to solve it.
 2. **Plan Your Solution:** Use flowcharts and pseudocode to outline your approach before coding.
 3. **Implement the Code:** Translate your pseudocode into the programming language of your choice.
 4. **Test Your Solution:** Run your code with various inputs to ensure its correctness and efficiency.
-

Flowchart and Pseudocode Examples

Flowchart

A flowchart is a diagram that represents the steps involved in solving a problem. It visually organizes the solution and helps in understanding the process flow.

Example:

- **Problem:** Sum of two numbers
- **Flowchart:**
 - Start
 - Input a & b
 - $sum = a + b$
 - Print sum
 - Exit

Pseudocode

Pseudocode is a plain language description of the steps in an algorithm. It helps in converting the problem-solving logic into actual code.

Example:

- **Problem:** Sum of two numbers
 - **Pseudocode:**
 1. Input a & b
 2. $\text{sum} = a + b$
 3. Print `sum`
 4. Exit
-

Sample Problems and Solutions

Problem 1: Area of a Square

- **Flowchart:**
 - Start
 - Input a
 - $\text{area} = a * a$
 - Print `area`
 - Exit
 - **Pseudocode:**
 1. Input a
 2. $\text{area} = a * a$
 3. Print `area`
 4. Exit
-

Problem 2: Minimum of Two Numbers

- **Flowchart:**
 - Start
 - Input a & b
 - Decision: $a < b$
 - Yes: Print a
 - No: Print b
 - Exit
- **Pseudocode:**
 1. Input a & b
 2. If $a < b$
 - Print a

Else

- Print b

3. Exit

Problem 3: Check if a Number is Odd or Even

- **Flowchart:**

- Start
- Input n
- Decision: $n \% 2 == 0$
 - Yes: Print Even
 - No: Print Odd
- Exit

- **Pseudocode:**

1. Input n
 2. If $n \% 2 == 0$
 - Print Even

Else

 - Print Odd
 3. Exit
-

Problem 4: Sum of Numbers from 1 to N

- **Flowchart:**

- Start
- Input n
- Initialize $sum = 0, count = 1$
- Loop: While $count \leq n$
 - $sum = sum + count$
 - Increment count
- Print sum
- Exit

- **Pseudocode:**

1. Input n
2. Initialize $count = 1, sum = 0$
3. While $count \leq n$
 - $sum = sum + count$
 - Increment count
4. Print sum

5. Exit

Problem 5: Check if a Number is Prime

- **Flowchart:**
 - Start
 - Input n
 - Initialize $i = 2$
 - Loop: While $i \leq n - 1$
 - Decision: $n \% i == 0$
 - Yes: Print Not Prime, Exit
 - No: Increment i
 - Print Prime
 - Exit
 - **Pseudocode:**
 1. Input n
 2. Initialize $i = 2$
 3. While $i \leq n - 1$
 - If $n \% i == 0$
 - Print Not Prime
 - Exit
 - Else
 - Increment i
 4. Print Prime
 5. Exit
-

Homework Problems

Q1: Calculate Simple Interest

- **Flowchart:**
 - Start
 - Input p, r, t
 - $\text{interest} = p * r * t$
 - Print interest
 - Exit
- **Pseudocode:**
 1. Input p, r, t
 2. $\text{interest} = p * r * t$
 3. Print interest
 4. Exit

Q2: Calculate Maximum of Two Numbers

- **Flowchart:**
 - Start
 - Input a & b
 - Decision: $a > b$
 - Yes: Print a
 - No: Print b
 - Exit
- **Pseudocode:**
 1. Input a & b
 2. If $a > b$
 - Print a
 - Else
 - Print b
 3. Exit

Q3: Calculate the Factorial of a Number

- **Flowchart:**
 - Start
 - Input n
 - Loop: While $n > 1$
 - $n = n * (n - 1)$
 - Decrement n
 - Print n
 - Exit
- **Pseudocode:**
 1. Input n
 2. While $n > 1$
 - $n = n * (n - 1)$
 - Decrement n
 3. Print n
 4. Exit

Q4: Determine Eligibility for a Driving License

- **Flowchart:**
 - Start
 - Input age
 - Decision: $\text{age} \geq 18$

- Yes: Print `You can drive`
 - No: Print `You cannot drive`
- Exit
- **Pseudocode:**
 1. Input `age`
 2. If `age >= 18`
 - Print `You can drive`
 - Else
 - Print `You cannot drive`
 3. Exit

How Does Code Run?

- **Editor or Software:** Write code in a code editor like Visual Studio Code.
- **Translation:** The code is translated by a compiler (e.g., C++ compiler) into machine language (01) that the computer can execute.

What to Install?

1. **Visual Studio Code:** A versatile code editor suitable for various programming languages.
2. **Set Up C++:**
 - **Windows:** Install MinGW or Visual Studio Community.
 - **Mac:** Install Xcode or use Homebrew to set up GCC.

Day 1 Notes

Prepared by Munawar Johar