Let’s Go with Algo

By Melakesleam Moges

14-10-2023

Know Your Teammate

ICE BREAKER

Ask the following Questions to you Partner and Remember them.

1. What is your Name?
2. What is you Mother's Name?
3. Do you have a pet? If so what is it?
4. What is the Name of your School?
5. What is your favorite game, sport or passtime activity?

BLAST FROM THE PAST

REVIEW

Problem Solving

1. What are the **four basic steps** to ***solving a problem***?
2. How can we break down a ***problem statement*** into its **core components**?
3. What are the **basic parameters** used to evaluate a ***possible solution*** to a problem?

Bonus:

On the “Scavenger Hunt” game we played, what did we learn about performing any set of tasks to reach a goal?

LECTURE 2

Understanding Algorithms

Understanding Algorithms - Objectives

To be able to

1. Explain what an algorithm is
2. Breakdown a solution/process into stages and steps
3. Writing the steps of algorithm
    a. Psudo code
    b. Flow chart

Understanding Algorithms - Basics

What is Algorithm?

Understanding Algorithms - Basics

According to www.merriam-webster.com

a procedure for solving a mathematical problem (as of finding the greatest common divisor) in a **finite number of steps** that frequently involves repetition of an operation

broadly : a **step-by-step** procedure for solving a problem or accomplishing some end

Understanding Algorithms - Basics

Exercise: Write down the algorithm for performing ONE of the following Tasks.

(Team Work)

1. Starting and Driving A Car
2. Setting up a GitHub account
3. Creating a repository on GitHub and cloning locally
4. Creating a branch from the main branch in a git repository and connecting it to the remote repo.
5. Finding the second largest number in a list of numbers.

- PRESENT TO CLASS -

Understanding Algorithms - Basic Problem Solving Strategies

There are different level of complexity to problems we face. Problems can range to simple to very difficult to those with no know solutions to date.

As the complexity increases, we need to utilize strategies and patterns to breakdown the problems to manageable & solvable size.

Understanding Algorithms - Basic Problem Solving Strategies

Common strategies are:

1. **Brute Force Strategy**
   - intuitive, direct, and straightforward approach in which all the possible ways are enumerated
   - Eg. exploring all the paths to a nearby market to find the minimum shortest path
   - A “trial and error” approach - non-optimal

Understanding Algorithms - Basic Problem Solving Strategies

Common strategies are:

1. Brute Force Strategy - **pros and cons**
    - **Pros**
        i. a guaranteed way to find the correct solution by listing all
        ii. a generic method - for all domain
        iii. for small and simpler problems
        iv. simple and good as a comparison benchmark

Understanding Algorithms - Basic Problem Solving Strategies

Common strategies are:

1. Brute Force Strategy - **pros and cons**
    - **Cons**
        i. Inefficient, especially for real-time problems
        ii. relies heavily on computational/processing power, rather than good algorithm design
        iii. Slow, time inefficient

Understanding Algorithms - Basic Problem Solving Strategies

Common strategies are:

1. **Divide and Conquer Strategy**
    - often involves a recursive approach
    - applying the same rule to divide a collection until you've broken it down into the smallest components and identify the answer.

Understanding Algorithms - Basic Problem Solving Strategies

Common strategies are:

1. Divide and Conquer Strategy - Approach
    - **Divide:** This involves dividing the problem into smaller sub-problems.
    - **Conquer:** Solve sub-problems by calling recursively until solved.
    - **Combine:** Combine the sub-problems to get the final solution of the whole problem.

Understanding Algorithms - Basic Problem Solving Strategies

Common strategies are:

1. Divide and Conquer Example - Binary Search
   - Performed on a sorted list of items based on a measurable property
   - Starting from the center of the list divide into two parts
   - Based on the search criteria, identify in which part the item could be in
   - Eliminate the other part and repeat the process until the item is found or proven it doesn't exist in the list.

Understanding Algorithms - Writing the Steps

Writing down an algorithm allows us to design, understand and enhance solutions to problems before we implement them. This “abstracts” the solution away from any implementation technologies.

What is the benefit of doing this?

There are two common approaches to writing algorithms: Pseudo code and Flow Chart

Understanding Algorithms - Writing the Steps

Pseudo code

- High-level, human readable, description of an algorithm
- More structured than english
- Less detailed than an implementation
- Hides the implementation related issues

Understanding Algorithms - Writing the Steps

Pseudo code - Example for Finding Max element in a List

```
Algorithm listMax(L, n)
    INPUT list L of n integers
    OUTPUT maximum element of L

    currentMax ← L[0]
    FOR i ←  1 TO n-1 DO
        IF L[i] > currentMax THEN
            currentMax ←  L[i]

    RETURN currentMax
```

```
Algorithm listMax(L, n)
    INPUT list L of n integers
    OUTPUT maximum element of L

    currentMax ASSIGN L[0]
    FOR each index of list L
        IF L[index] > currentMax THEN
            currentMax ASSIGN L[index]

    RETURN currentMax
```

Understanding Algorithms - Writing the Steps

Pseudo code - common syntax

Assigning value to variable/container in memory - initializing

container ← value

Conditional selections

```
IF condition THEN
    … tasks …
ELSE
    … tasks …
```

Understanding Algorithms - Writing the Steps

Pseudo code - common syntax

Looping through a set of tasks

FOR i ← 1 **TO** n-1 **DO**
    … tasks …

WHILE condition **DO**
    … tasks …

REPEAT
    … tasks …
UNTIL condition

Understanding Algorithms - Writing the Steps

Pseudo code - common syntax

Method declaration

Algorithm methodName(*parameters passed to it*)
INPUT define prime input types
OUTPUT define prime output types

Concluding the steps of the algorithm by returning or not an expected value

return [value]

Understanding Algorithms - Writing the Steps

Exercise(Pseudo code): Write pseudo code for Binary Search algorithm.

(team work) [10min]

Sample input list: [8 , 12 , 15 , 21 , 45]

Search for element: 12

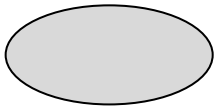
Understanding Algorithms - Writing the Steps

Flow Chart

- A graphical representation of an algorithm
- Represents the flow of control in program.
- Utilizes simple symbols to represent different action/stages
- Symbols are connected to represent flow of activity.

Understanding Algorithms - Writing the Steps

Flow Chart - Basic symbols



Terminal

indicates Start, Stop and Halt in a program's logic flow.

Input/Output

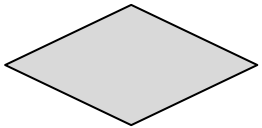
indicates instructions that take input from input devices and display output on output devices

Processing

indicates arithmetic processes such as adding, subtracting, multiplication and division

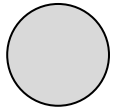
Understanding Algorithms - Writing the Steps

Flow Chart - Basic symbols



Decision

indicates operations such as yes/no question or true/false

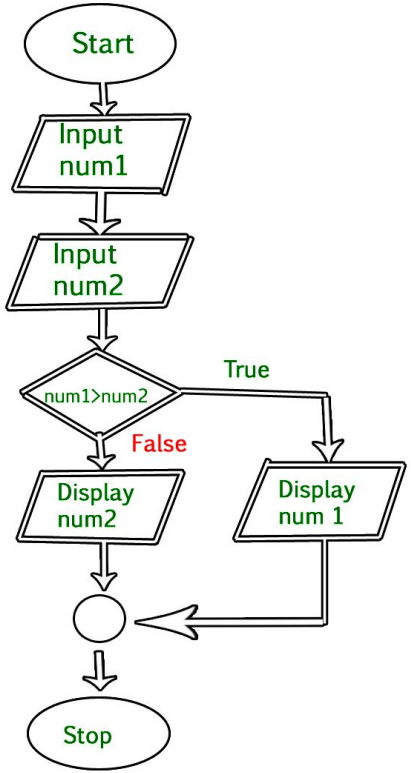


Connectors

Whenever flowchart becomes complex or it spreads over more than one page, connectors are used to avoid any confusions.

Understanding Algorithms - Writing the Steps

Flow Chart - Example



Understanding Algorithms - Writing the Steps

Exercise(Flow Chart): Write a flow chart of an algorithm to calculate two numbers provided by the user and display back their sum. You can use any graphics tools you know. Recommended: Google Slides, Visio,...

(team work) [10min]

Understanding Algorithms - Writing the Steps

PRACTICE WRITING AS MAY ALGORITHMS AS YOU CAN USING THESE TWO TECHNIQUES TO MASTER IT.

- END OF LECTURE -