An 8085 Microprocessor Simulator

- A presentation by Shrijan Regmi

About the app

Packed with a complete set of opcodes for your next 8085 Microprocessor program, the app is made using Flutter framework that supports Android, Web and Windows platforms.

Available Opcodes

- MVI, LXI, MOV, ADD, ADI, SUB, SUI, INR, INX, DCR, DCX, LDA, LDAX, STA, STAX, RST 5

1. Use of DSA

DSA is the building block of any software development. DSA is incorporated in the app as follows:

- **Array**
  Array of buttons, Array of Opcodes
- **Stack**
  Stack of memory address
- **Sorting**
  Sorting of the opcodes in ascending order
- **Searching**
  Searching of opcodes

Array

An array is a linear data structure that collects elements of the same data type and stores them in contiguous and adjacent memory locations. Arrays work on an index system starting from 0 to (n-1), where n is the size of the array.

Why Do You Need an Array in Data Structures?

Let's suppose a class consists of ten students, and the class has to publish their results. If you had declared all ten variables individually, it would be challenging to manipulate and maintain the data.

If more students were to join, it would become more difficult to declare all the variables and keep track of it. To overcome this problem, arrays came into the picture.

Example of Array in Dart

```dart
List<String> _listOfFruits = ['apple', 'orange', 'banana'];
```

Stack

Stacks in Data Structures is a linear type of data structure that follows the LIFO (Last-In-First-Out) principle and allows insertion and deletion operations from one end of the stack data structure, that is top. Implementation of the stack can be done by contiguous memory which is an array, and non-contiguous memory which is a linked list. Stack plays a vital role in many applications.

Types of implementation of Stack

- Static Implementation (Array Implementation)
- Dynamic Implementation (Linked List Implementation)

Example of Stack in Dart

```dart
List<int> _stack = [1, 2, 3];

void _push() {
  _stack.add(4);
}

void _pop() {
  _stack.removeLast();
}
```

Sorting

Sorting is the process of arranging items in a specific order or sequence. It is a common algorithmic problem in computer science and is used in various applications such as searching, data analysis, and information retrieval.

In other words, you can say that sorting is also used to represent data in a more readable format.

Common types of Sorting

- Bubble Sort,
- Insertion Sort,
- Selection Sort,
- Merge Sort,
- Shell Sort etc

Example of Sorting in Dart

```
List<String> _listOfFruits = ['apple', 'orange', 'banana'];

_listOfFruits.sort(); // [ apple, banana, orange ]
```

Searching

Searching in data structure refers to the process of finding the required information from a collection of items stored as elements in the computer memory. These sets of items are in different forms, such as an array, linked list, graph, or tree. Another way to define searching in the data structures is by locating the desired element of specific characteristics in a collection of items.

Common types of Searching

- Sequential Search,
- Binary Search

Example of Searching in Dart

```
List<String> _listOfFruits = ['apple', 'orange', 'banana'];
String _searchText = "orange";

void _searchOrange() {
  List<String> _result = _listOfFruits
          .where((element) => element == _searchText)
          .toList(); // [orange]
}
```

Conclusion, Data structure

and algorithms is a branch of computer science that deals with creating machine-efficient and optimized computer programs. The term refers to the storage and organization of data, and Algorithm refers to the step by step procedure to solve a problem.