Lesson 3: Data Structures; Problem Solving Techniques

Aniruddha Kudalkar

Objective

- Psychology of Problem Solving
- Algorithms
- Data Structures
- Few Problems

Psychology Of Problem Solving

Psychology

How our brain looks at problem and tell us solution

- Heuristics
- Trial-and-Error
- Insights
- Algorithms

Heuristics

Mental strategy based on rule-of thumb. There is no guarantee that it will always work out to produce the best solution. Simplifies complex problems by narrowing the possible solutions.It makes it easier to reach the correct solution using other strategies.

Heuristics

Lets buy a Stock

Trial and Error

Trying a number of different solutions and ruling out the ones that do not work. It can be highly time-consuming. If you are near to reached solution, it is best strategy.

Insights

Insight is something that just occurs suddenly. Researchers suggest that insight can occur if you’ve dealt with similar problems in the past.

Insights

Solving algebraic/physics equations.

Algorithms

The step-by-step procedure involved in figuring out the correct answer to any problem is called algorithm. Algorithm is the strategy that results in accurate answer; however, it’s not always practical. The strategy is highly time consuming, and involves taking lots of steps.

Algorithms

The step by step procedure involved in solving a mathematical problem using math formula is a perfect example of a problem-solving algorithm.

Algorithms

Algorithm

How you build the algorithm exactly ?

- Design Techniques
- Algorithmic Efficiency

Design Techniques

technique is very important, before you start designing

Design Techniques

- Brute Force
- Divide and Conquer
- Greedy Algorithms
- Dynamic Programming

Brute Force

brute-force search or exhaustive search, also known as generate and test, is a very general problem-solving technique, which consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem statement.

Brute Force

A brute-force algorithm that finds the number n is prime or not.

Divide and Conquer

recursively breaks down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem.

Divide and Conquer

Searching number n in sorted array

Greedy Algorithms

algorithm that follows the problem-solving heuristic of making the locally optimal choice at each stage. A greedy strategy does not produce an optimal solution, but a can yield locally optimal solutions that approximate a globally optimal solution in a reasonable amount of time.

Greedy Algorithms

determine the minimum number of coins to give while making change.

Dynamic Programming

simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner. Problem can be solved optimally by breaking it into sub-problems and then recursively finding the optimal solutions to the sub-problems.

Dynamic Programming

Shortest Path Algorithm

Algorithmic Efficiency

In computer science efficiency is very important

Algorithmic Efficiency

is a property of an algorithm which relates to the amount of computational resources used by the algorithm. It can be a thought of as analogous to engineering productivity for a repeating or continuous process.

maximum efficiency = minimize resource usage.

Big O Notation

The most commonly used notation to describe resource consumption or "complexity" is Big O notation, representing the complexity of an algorithm as a function of the size of the input.

$f(n) = O(n)$

Big O Notation

- O(1) - Constant - Finding the median from a sorted list
- O(log n) - Logarithmic - Binary Search
- O(n) - Linear - Searching in unsorted array
- O(n log n) - linearithmic - merge sort
- O(n^2) - Quadratic - n digit multiplication algorithm
- O(c^n) - Exponential - check two statements are equivalent

Measures of resource usage

- Time: how long does the algorithm take to complete?
- Space: how much working memory (typically RAM)
- Direct power consumption: power needed directly
- Indirect power consumption: for cooling, light etc
- Transmission Size, External Space, Response Time, Ownership Cost

Data Structures

Data Structures

Store, Search and Sort

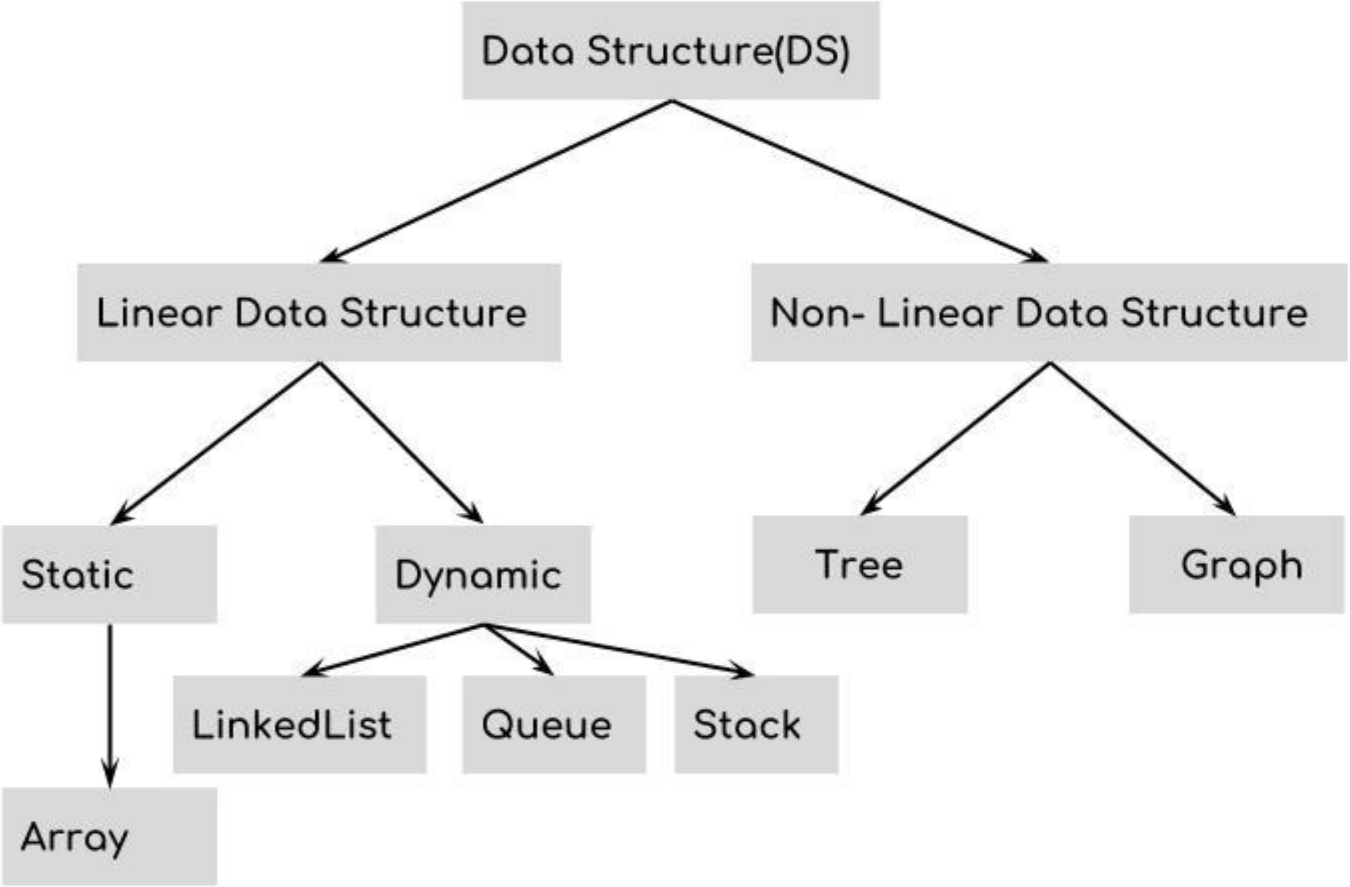
- DS Basics
- Searching Algorithms
- Sorting Algorithms

DS Basics

Basic understanding of data structures

Data Structure

a data structure is a data organization, management, and storage format that enables efficient access and modification. More precisely, a data structure is a collection of data values, the relationships among them, and the functions or operations that can be applied to the data.


Data Structure(DS)
Linear Data Structure
Non- Linear Data Structure
Static
Dynamic
Tree
Graph
LinkedList
Queue
Stack
Array

Linear Data Structure

the elements are arranged in sequence one after the other. Since elements are arranged in particular order, they are easy to implement.

However, when the complexity of the program increases, the linear data structures might not be the best choice because of operational complexities.

Linear Data Structure

- Array
- Stack
- Queue
- LinkedList

Non Linear Data Structure

elements in non-linear data structures are not in any sequence. Instead they are arranged in a hierarchical manner where one element will be connected to one or more elements.

Non-linear data structures are further divided into graph and tree based data structures.

Non Linear Data Structure

- Graph
- Tree
- Binary Tree

Useful Data Structures

- Hashtable
- Heap
- Set

etc

Searching Algorithms

Different searching techniques

Searching Algorithms

Searching Algorithms are designed to check for an element or retrieve an element from any data structure where it is stored.

Based on the type of search operation, these algorithms are generally classified into two categories i.e. Sequential Search and Interval Search

Sequential Search

In this, the list or array is traversed sequentially and every element is checked.

Example : Linear Search

Interval Search

These algorithms are specifically designed for searching in sorted data-structures. These type of searching algorithms are much more efficient than Linear Search as they repeatedly target the center of the search structure and divide the search space in half.

Example : Binary Search

Searching Algorithms

- Linear Search
- Binary Search
- Jump Search
- Interpolation Search
- Exponential Search
- Sublist Search
- Fibonacci Search
- Ubiquitous Binary Search

etc

Sorting Algorithms

Different sorting techniques

Sorting Algorithms

used to rearrange a given array or list elements according to a comparison operator on the elements. The comparison operator is used to decide the new order of element in the respective data structure.

Sorting Algorithms

- Selection Sort
- Bubble Sort
- Insertion Sort
- Bucket Sort
- Radix Sort
- Heap Sort
- Quick Sort
- Merge Sort
- Recursive Insertion Sort
- Shell Sort

etc

Few Problems

Lets Try

- Find a Fixed Point - Linear Search
- Reverse A Linked List
- Sorting Strings using Bubble Sort

Thanks,

LEARN, CODE, EARN

Credits

- https://en.wikipedia.org/
- https://docs.oracle.com/