# Algorithms & Data Structures

In this week we will look at the core data structures and algorithms used in everyday applications. We will discuss the trade-offs involved with choosing each data structure, and each algorithm, along space and time complexity.

### **Day 1 – Office/Zoom**

We will talk about basic algorithms and data structures, time and space complexity, and understand why algorithms are so very importance for every software engineer

### Data Structures

* Array
* Linked List
* Hash Table
* Tree
  + BST
  + Trie
  + Heap
* Queue
* Stack
* Graph
  + Graph representations

### Complexity

* Big O Notation
* Importance of Big O Notation
* Time & Space complexity
* Trade-offs

### Algorithms

* What is an algorithm
* Day to day usages
* Common algorithms
  + Sorting
  + Shortest Path
  + Minimal Spanning Tree
  + BSA
  + BFS & DFS
  + Topological sort
  + Flood fill

### Home Assignments

* Walkthrough the materials and assignments, which will consist implementing some basic algorithms and solve basic problems using a specific data structure

### **Day 1 – Home**

This day will focus on learning basic data structures and practice the use of them.

You will read some materials and then solve some problems.

Some of the information is accumulative and will be needed for the next days, and some is basic knowledge that will be needed as a software engineer, either way, you are expected to read all the materials.

Following the reading materials, you will have a variety of exercises, try to solve each before jumping to the solution

### Data Structures

### Hash

* + Intro - <https://www.geeksforgeeks.org/hashing-set-1-introduction/>
  + Separate Chaining for Collision Handling - <https://www.geeksforgeeks.org/hashing-set-2-separate-chaining/>
  + Open Addressing for Collision Handling - <https://www.geeksforgeeks.org/hashing-set-3-open-addressing/>
  + Double Hashing - <https://www.geeksforgeeks.org/double-hashing/>
  + Exercises
    - <https://www.geeksforgeeks.org/find-top-three-repeated-array/>
    - <https://www.geeksforgeeks.org/check-if-two-arrays-are-equal-or-not/>
    - <https://www.geeksforgeeks.org/sum-fai-aj-pairs-array-n-integers/>
    - <https://www.geeksforgeeks.org/count-index-pairs-equal-elements-array/>

### Graph

* + Graph and its representations - <https://www.geeksforgeeks.org/graph-and-its-representations/>
  + Breadth First Traversal for a Graph - <https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/>
  + Depth First Traversal for a Graph - <https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/>
  + Applications of Depth First Search - <https://www.geeksforgeeks.org/applications-of-depth-first-search/>
  + Applications of Breadth First Traversal - <https://www.geeksforgeeks.org/applications-of-breadth-first-traversal/>
  + Exercises
    - <https://www.geeksforgeeks.org/stepping-numbers/>
    - <https://www.geeksforgeeks.org/count-number-trees-forest/>

### Binary Tree

* + Intro - <https://www.geeksforgeeks.org/binary-tree-set-1-introduction/>
  + Properties - <https://www.geeksforgeeks.org/binary-tree-set-2-properties/>
  + Types - <https://www.geeksforgeeks.org/binary-tree-set-3-types-of-binary-tree/>
  + BFS vs DFS for Binary Tree - <https://www.geeksforgeeks.org/bfs-vs-dfs-binary-tree/>
  + Array implementation of Binary Tree - <https://www.geeksforgeeks.org/binary-tree-array-implementation/>
  + Exercises
    - <https://www.geeksforgeeks.org/find-n-th-node-inorder-traversal/>
    - <https://www.geeksforgeeks.org/check-leaves-level/>

### **Day 2 – Home**

This day will focus on taking our very basic daily usage of algorithms, and implement them by ourselves. You are expected to understand the implementation of the code in Pseudo code, and implement it in Java. Afterwards, more very interesting useful algorithms to read about.

Some sort algorithms visualizations:

* + <https://www.youtube.com/watch?v=TZRWRjq2CAg&ab_channel=udiprod>
  + <https://www.youtube.com/watch?v=aXXWXz5rF64&ab_channel=udiprod>
  + <https://www.youtube.com/watch?v=lyZQPjUT5B4&ab_channel=AlgoRythmics>
  + <https://www.youtube.com/watch?v=es2T6KY45cA&ab_channel=udiprod>
  + I added - https://www.youtube.com/watch?v=bfzYj-qGw7U

### Algorithm’s implementations

1. **Bubble sort** - <https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm>
2. **Merge sort** - <https://www.tutorialspoint.com/data_structures_algorithms/merge_sort_algorithm.htm>
3. **BSA** - <https://www.geeksforgeeks.org/binary-search/>

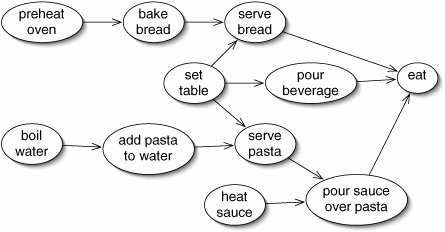
### Reading materials

* Minimum Spanning tree - <https://www.tutorialspoint.com/data_structures_algorithms/spanning_tree.htm>
  + Kruskal - <https://www.tutorialspoint.com/data_structures_algorithms/kruskals_spanning_tree_algorithm.htm>
  + K Centers Problem - <https://www.geeksforgeeks.org/k-centers-problem-set-1-greedy-approximate-algorithm/>

### **Day 3 – Home**

The interesting part – Graph algorithms.

We will take a situation from real life, convert our problem into graph, and the implement some very basic and important algorithms. Later on, we will review more real-life problems and their solutions using graphs.



Graph representation - A graph can be represented in several ways; we will take a photo and convert it to 2 representations of a graph. The first will be **Adjacency list**, and the second, **Adjacency Matrix**.

1. <https://www.geeksforgeeks.org/comparison-between-adjacency-list-and-adjacency-matrix-representation-of-graph/>

Graph Traversal – there are 2 main ways to iterate over a Graph, and we will implement both of them - <https://www.tutorialspoint.com/difference-between-bfs-and-dfs>

1. **BFS**
2. **DFS**

Topological Sorting – One of the basic and very often forgotten algorithm is the ability to topological sort things via a Graph; We will implement Topological sorting for DAG (Directed Acyclic Graph), and do a small research of its real-life usages

1. **Topological Sorting** - <https://www.hackerearth.com/practice/algorithms/graphs/topological-sort/tutorial/>
2. Exercise (Research) – find 5 usages of Topological Sorting Algorithm in real life

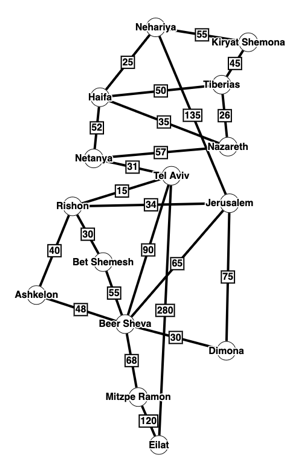
Yet more problems we can easily solve with a Graph

* Snake and Ladder Problem - <https://www.geeksforgeeks.org/snake-ladder-problem-2/>
* Detect Cycle in a Directed Graph – <https://www.geeksforgeeks.org/detect-cycle-in-a-graph/>

### **Day 4 – Home**

This day will be dedicated solely to one very important algorithm – Dijkstra.

We have many (!) usages of Shortest Path algorithms, today we will implement one of them.

1. Convert a photo to 2 representations of a Graph (use your functions from yesterday)
2. Read and understand how Dijkstra algorithm works, then implement it in Java - <https://www.freecodecamp.org/news/dijkstras-shortest-path-algorithm-visual-introduction/>

There are other Shortest Path algorithms, and it’s important to understand their usages and trade-offs

* Bellman Ford - <https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/>
* Floyd Warshall - <https://www.geeksforgeeks.org/floyd-warshall-algorithm-dp-16/>
* A Star - <https://www.geeksforgeeks.org/a-search-algorithm/>

### **Day 5 – Office/Home**

Last but not least, this day will be dedicated to broaden our knowledge, with various of developers and lecturers.

From morning to noon, you will participate some very interesting sessions

* Summary of our intensive week and home assignments
* Industry practical day-to-day work – from logs to graphs
* Map’s navigation – dive in to shortest path algorithm
* Use case of a complicated algorithm usages in Big Data environment

הי,

הקישור לריפוזיטורי שלי-

אני רוצה ואשמח לקבל הערות/הארות אם צריך לייעל את הפתרונות

קצת על התרגילים:

בגרפים:

בתחילה ניסיתי לממש בעצמי את הגרף (סיפור שלקח לי כמה שעות ולא בדיוק הצליח...) אבל הייתי בכיוון:),

בעץ: השתמשתי בפרויקט קיים של עץ שמימשתי במהלך הלימודים ובו בניתי את הפונקציה בסוף הדף של פרוגרם.

תודה רבה על כל ההשקעה הזאת...

זה באמת ממריץ להשקיע גם😊