echo "# Datastructures\_and\_algorithms" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/seppotk/Datastructures\_and\_algorithms.git

git push -u origin main

**…or push an existing repository from the command line**

git remote add origin https://github.com/seppotk/Datastructures\_and\_algorithms.git

git branch -M main

git push -u origin main

<https://github.com/TT00FE39-3001/lecture1>

<https://cpp.sh/>

**Outline**

* Data structures & Abstract Data Types (ADT)
* Algorithms
* Course Mechanics?

**Part 1**

* [Abstract Data Types (ADT)](https://en.wikipedia.org/wiki/Abstract_data_type)
* [Data Structures](https://en.wikipedia.org/wiki/Data_type)
* Data structures are language dependant

[**Data Structures In C++**](https://www.softwaretestinghelp.com/data-structures-in-cpp/)

* Introduction
* Classification
* [Advantages](https://www.softwaretestinghelp.com/data-structures-in-cpp/)

**Stacks vs Queues (1/2)**

* [Stack Data Structure In C++ With ARRAY/Linked list implementation](https://www.softwaretestinghelp.com/stack-in-cpp/)
* [Queue Data Structure In C++ With Illustration](https://www.softwaretestinghelp.com/queue-in-cpp/)

**Stacks vs Queues (2/2)**

* [Data structures 101: Stacks vs Queues](https://www.educative.io/blog/data-structures-stack-queue-java-tutorial)
* [Visualization](https://www.cs.usfca.edu/~galles/visualization/Algorithms.html)
* [FIFO](https://en.wikipedia.org/wiki/FIFO_(computing_and_electronics))
* [LIFO](https://en.wikipedia.org/wiki/Stack_(abstract_data_type))

**Applications**

* Recursion
* Browser history
* Undo (Word processor)

**Part 2**

* Classification
* Analysis of Algorithm Efficiency
* Space vs Time
* Algorithms language agnostic
  + Pseudo code / C++ etc
* [Introduction To Searching Algorithms In C++](https://www.softwaretestinghelp.com/searching-algorithms-in-cpp/)
* [Introduction To Sorting Techniques In C++](https://www.softwaretestinghelp.com/sorting-techniques-in-cpp/)
* [Algorithms In STL](https://www.softwaretestinghelp.com/algorithms-in-stl/)

**Classification**

* Brute Force
* Divide-and-Conquer
* Transform-and-Conquer
* Greedy Technique
* Dynamic Programming

**Analysis of Algorithm Efficiency**

[Big-O, Little-O, Theta, Omega](https://cathyatseneca.gitbooks.io/data-structures-and-algorithms/content/analysis/notations.html)

* Big �, Little �
* Big Ω, Little �
* Θ

**Space vs Time**

* Same tools
* Space and Time Trade-Offs

**Tools**

* [Algorithm Visualizations 1](https://www.youtube.com/playlist?list=PLlzjt-kuLwlo6QuXHSZCkITkfn9M2yHg1)
* [Algorithm Visualizations 2](https://gbhat.com/algorithms/selection_sort.html)
* [Animation engine for explanatory math videos](https://github.com/3b1b/manim)

#include <iostream>

using namespace std;

int main()

{

cout<<"Hello,World!! This is C++ Tutorial!!\n";

cin.get();

return 0;

}