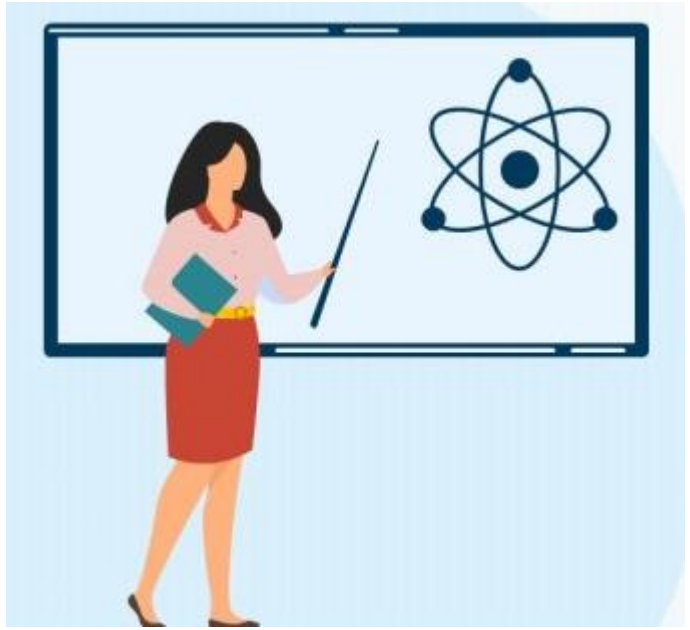
Introduction to Algorithm EC2206



Welcome

2024.8

Instructor

Suman Pandey

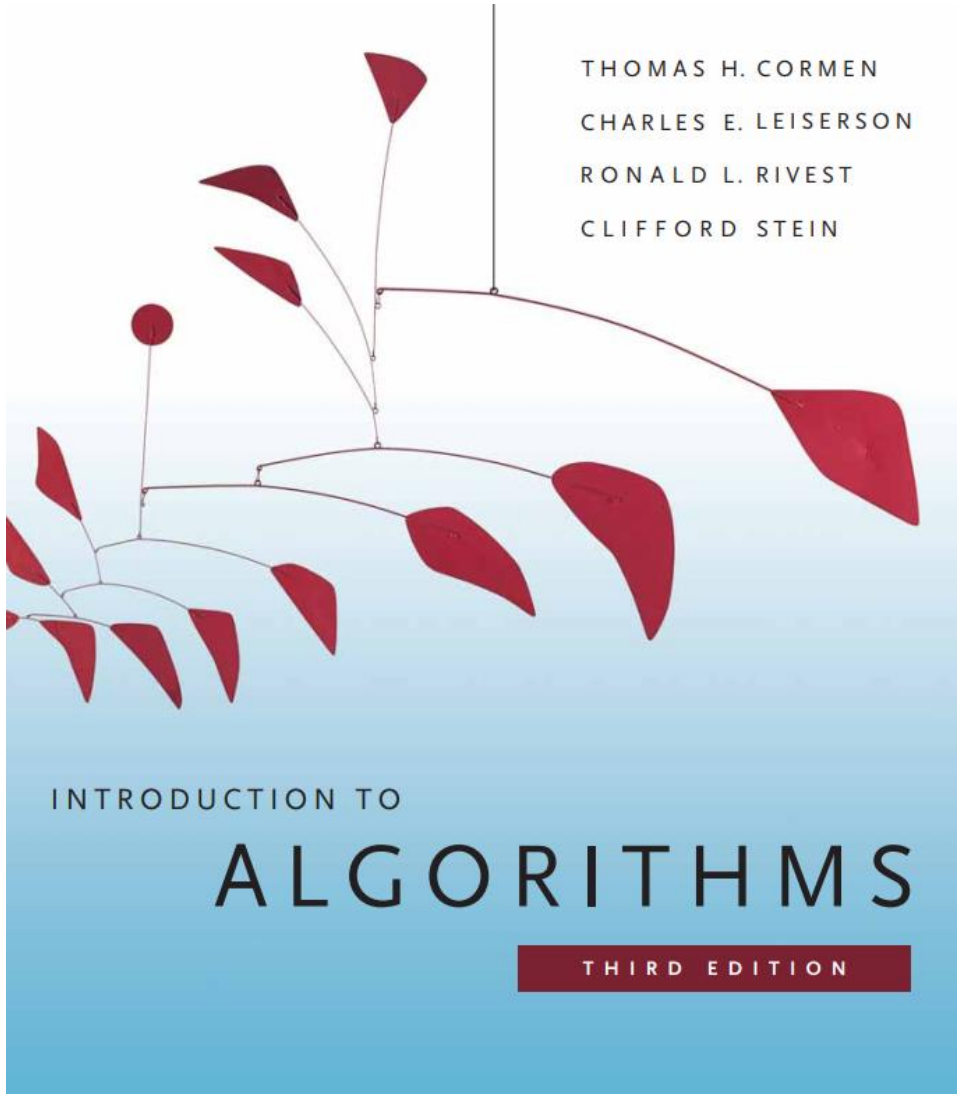
Conduct – Online
Email – suman17july@gmail.com, suman@postech.ac.kr
Office – EECS C3, C wing, 205 room
Phone – 010-3742-1791

Technical Assistant

Fullname	Email address	Mobile phone
Seunghyuck Hyun	hyun_sh_ug@gm.gist.ac.kr	010-3156-5525
Lim, Chaehyun (임채현)	bkindtoevery1@gm.gist.ac.kr	010-4386-0909
Lee, Jae hee (이재희)	jhlee.ug@gm.gist.ac.kr	010-9177-0913
Altynbek kyzy,Sumaiia	sumaiiaaltyn@gm.gist.ac.kr	010-5083-8220

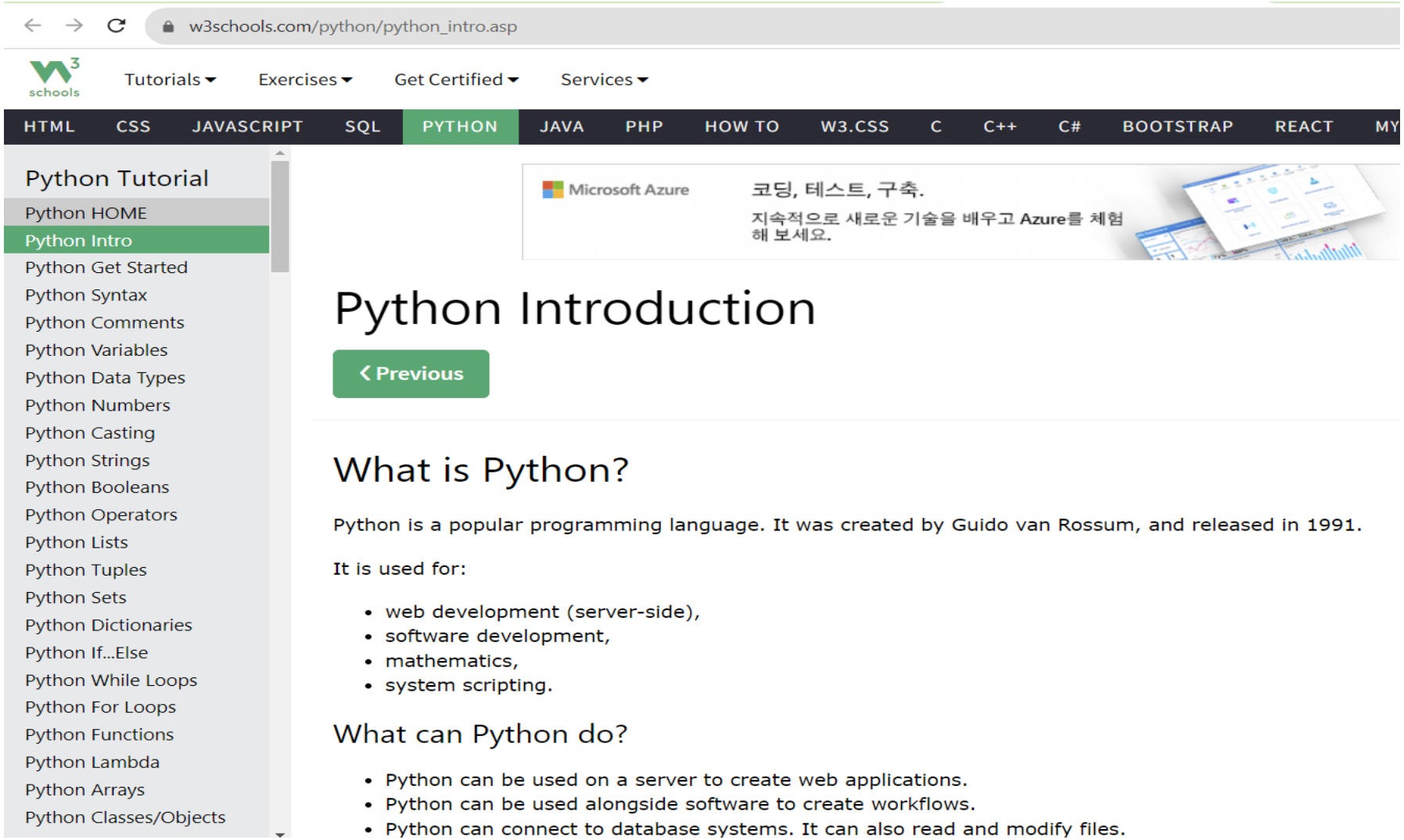
- **Assignments and Grading** – Technical Assistants will be responsible to clarify your doubts about Assignments and Grading

Book


THOMAS H. CORMEN
CHARLES E. LEISERSON
RONALD L. RIVEST
CLIFFORD STEIN
INTRODUCTION TO
ALGORITHMS
THIRD EDITION

Python Basics

- https://www.w3schools.com/python/python_intro.asp



Why use Python for Algorithm

- Once your learn AL, you can proceed with AI easily using python

Python Installation

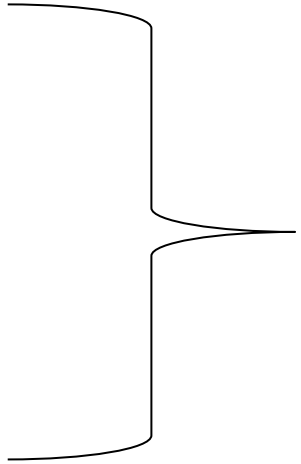
- **Install Anaconda**
  - https://docs.anaconda.com/anaconda/user-guide/getting-started/#nav-hello
  - **Use Jupiter notebook to write your code**

Agenda

- Role of Algorithms
- Big O
- Recursion, Divide and Conquer
- Sorting Algorithms
- Dynamic Programming
- Matrix-Chain multiplication
- Greedy Algorithm
- **Mid Term**
- Binary Trees / Heap
- Graph
- Minimum Spanning Tree
- Shortest Path / Topological Sort
- **Final Term**
- **If we have time**
- Integer Linear Programing
- NP-Completeness

How Much Coding you will do ?

- **Assignment -** There will be several coding assignments during this course
  - Basic Python Skills – 5 question (easy)
  - Sorting – 2 questions (easy)
  - Recursion – 2 question (medium)
  - Dynamic Programming – 3 questions (easy , medium, hard)
  - Greedy – 2 question (easy, medium)
  - Graph – 2 question (medium, hard)



30% of your grades

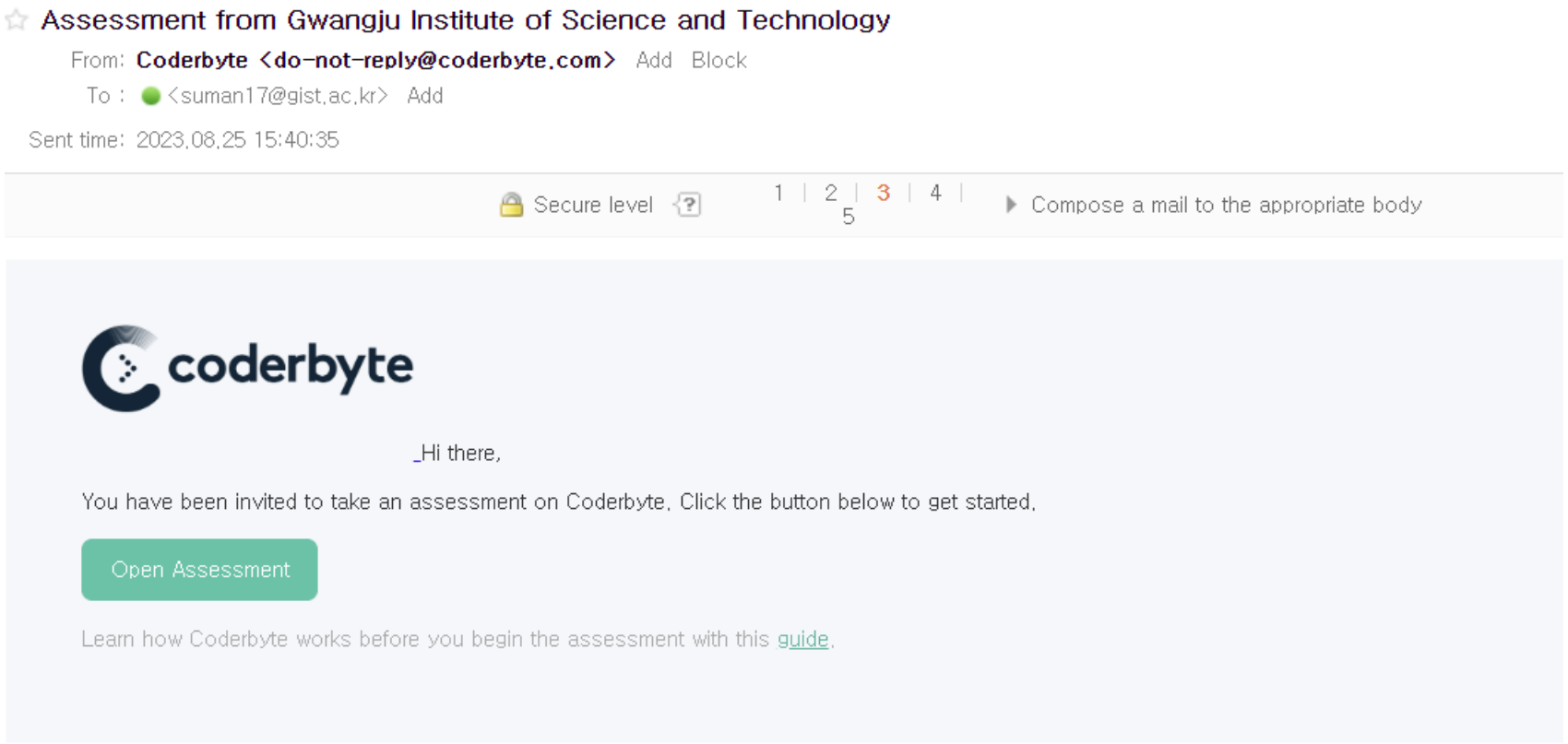
- **Complexity of Assignment** -
  - Easy
  - Medium
  - Hard
- **Queries related to coding should be directed to Technical Assistants**
- **I will show you sample code for algorithms as we go through each of them.**

Method for Assignment Submission

- Auto graded challenges – Through out the course 10 different challenges
- Tool used – Coderbyte.com
- The tool will auto grade your submission
- However, TAs, will make sure if the submitted assignment has best time complexity

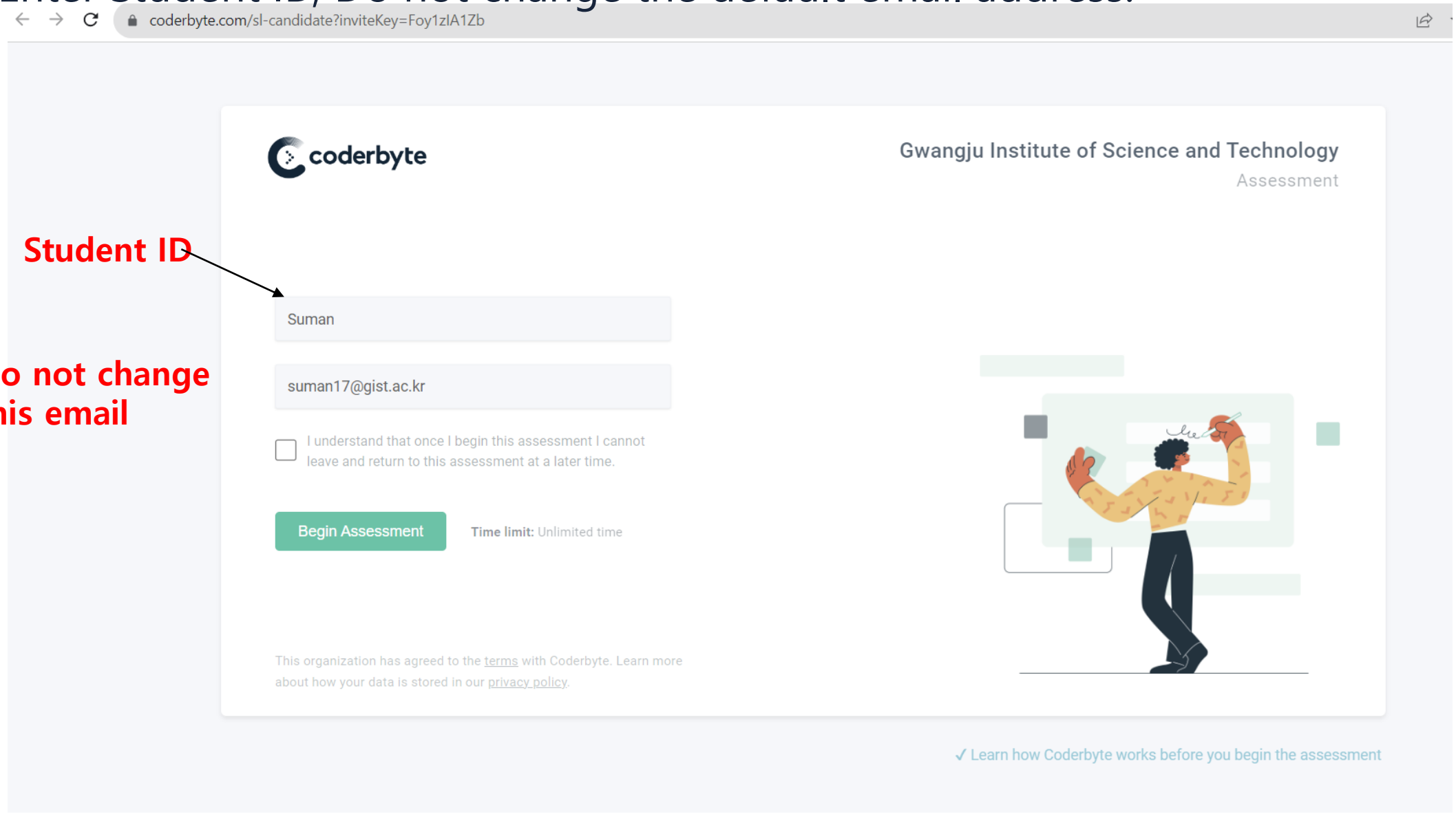
Assignment Format

- You will receive an email for each assignment.
- Open this assignment, when you have enough time and energy to solve the challenge. Going back and forth with these assignment is not possible.



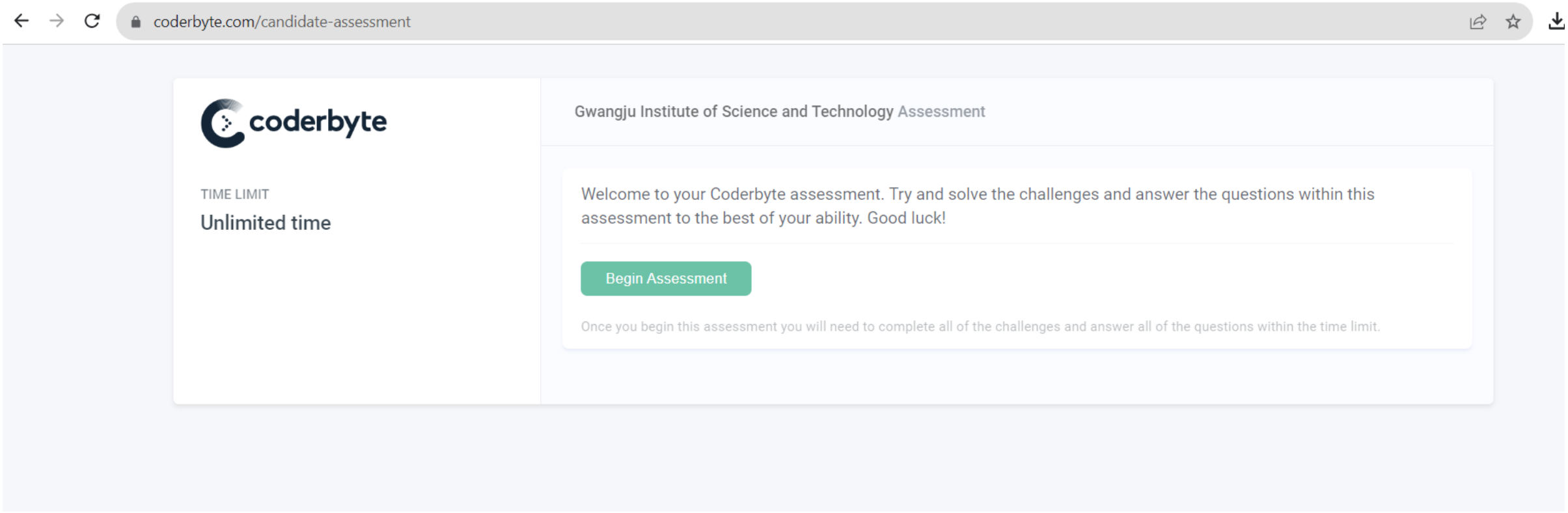
Please Enter Student ID here

- Enter Student ID, Do not change the default email address.



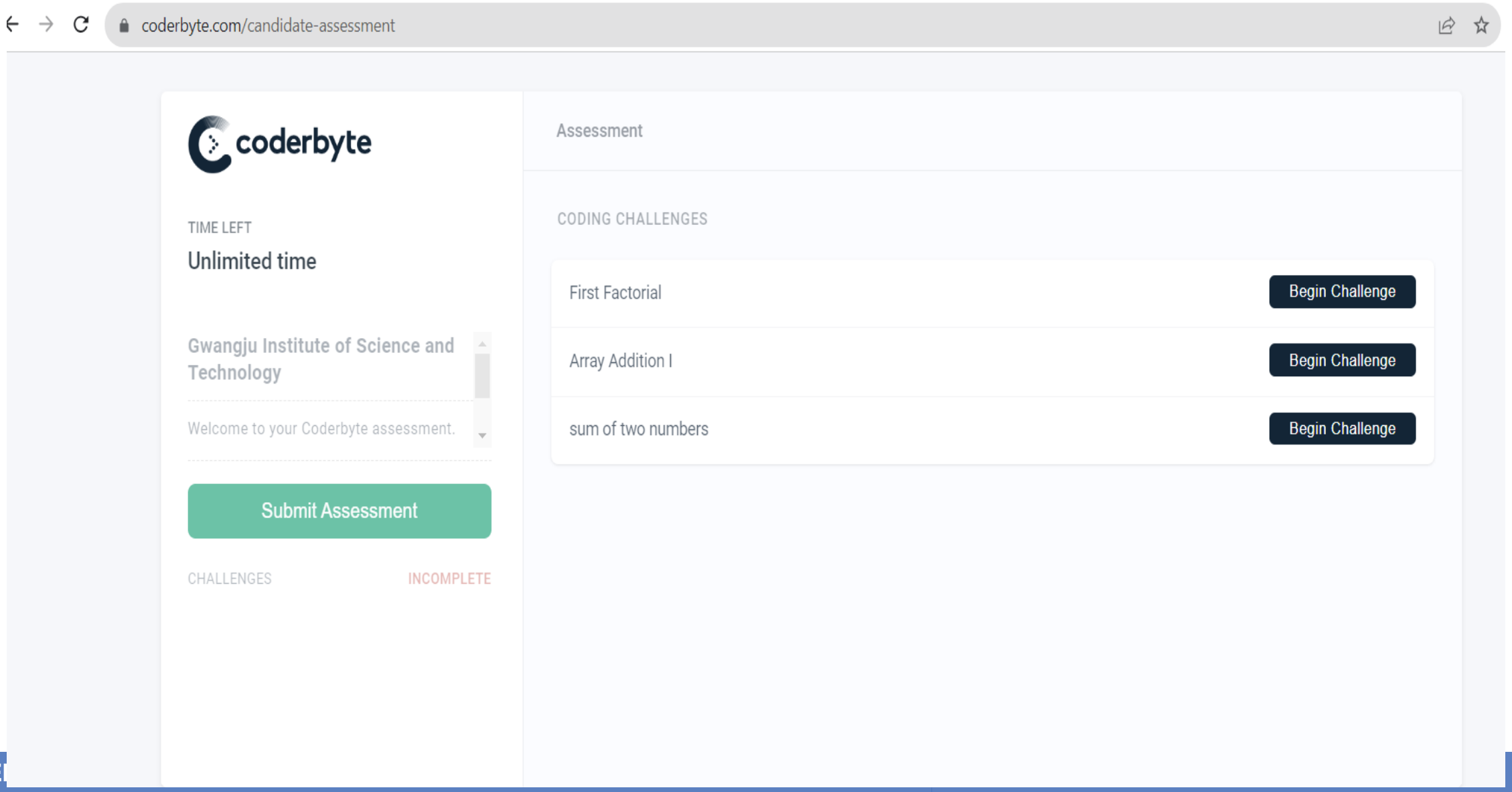
Begin Submitting Assignment

- Begin the assignment, Mostly you will get entire weekend to solve your assignment. Unless I take a mock test, where I can limit the time.

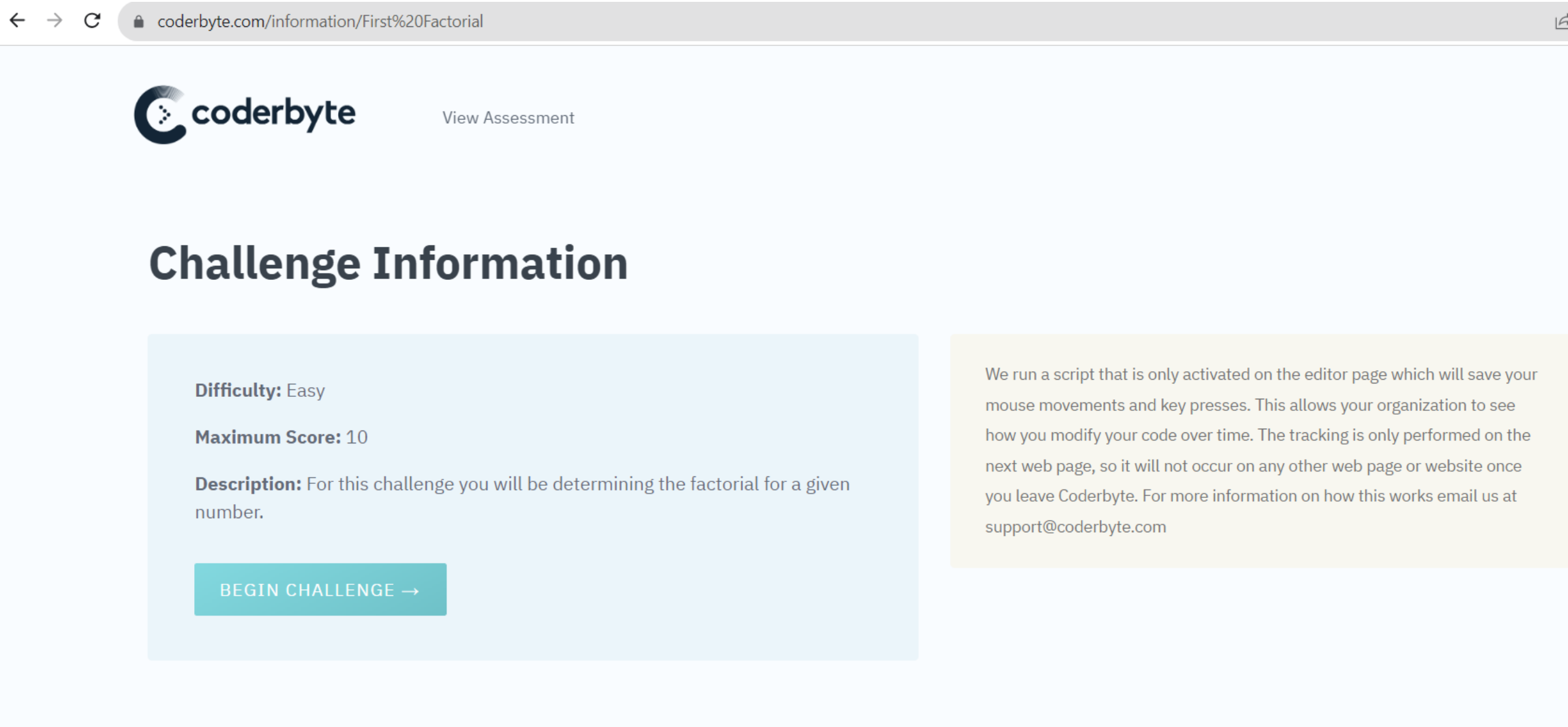


Click on Begin Challenge

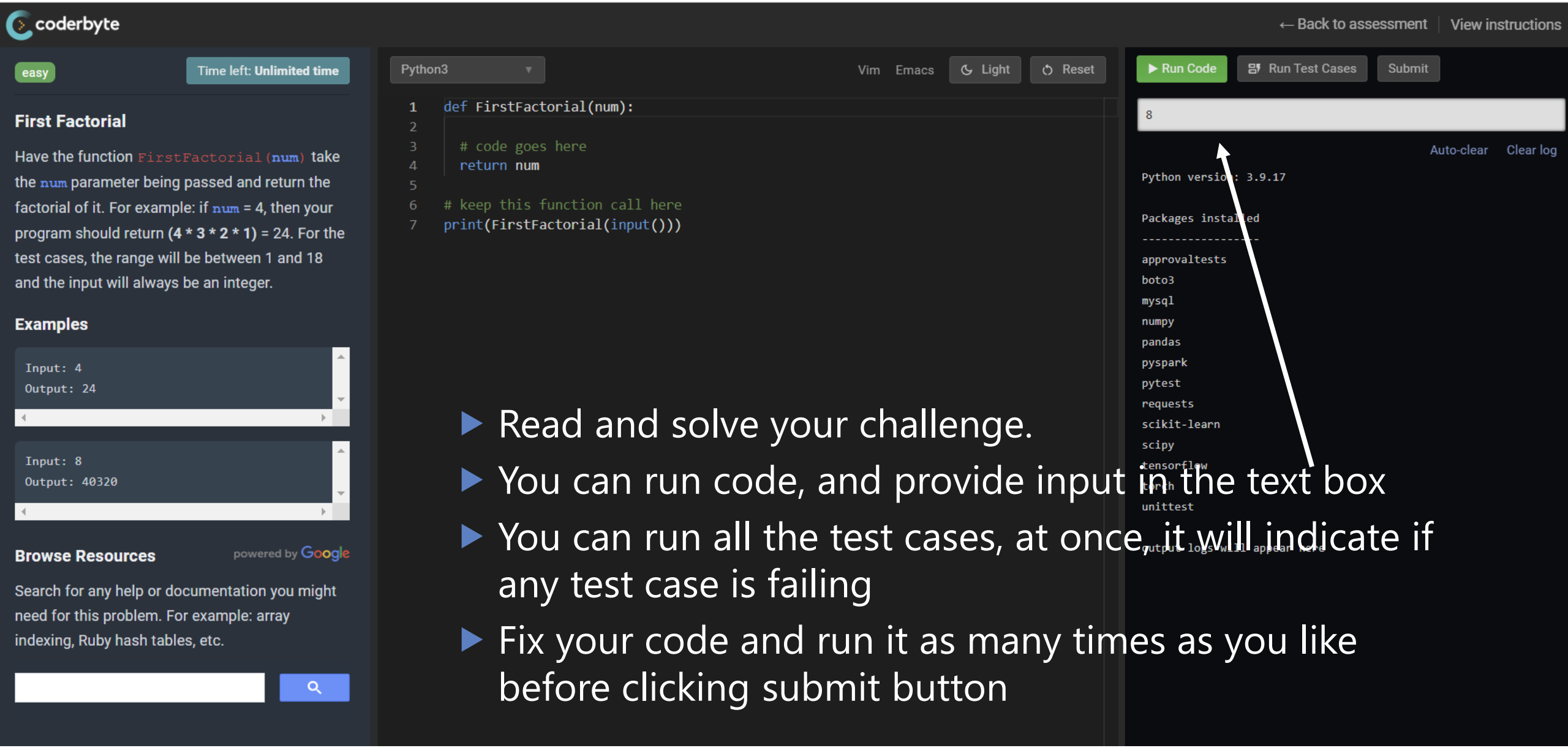
- Click on Begin Challenge.



Click on Begin Challenge



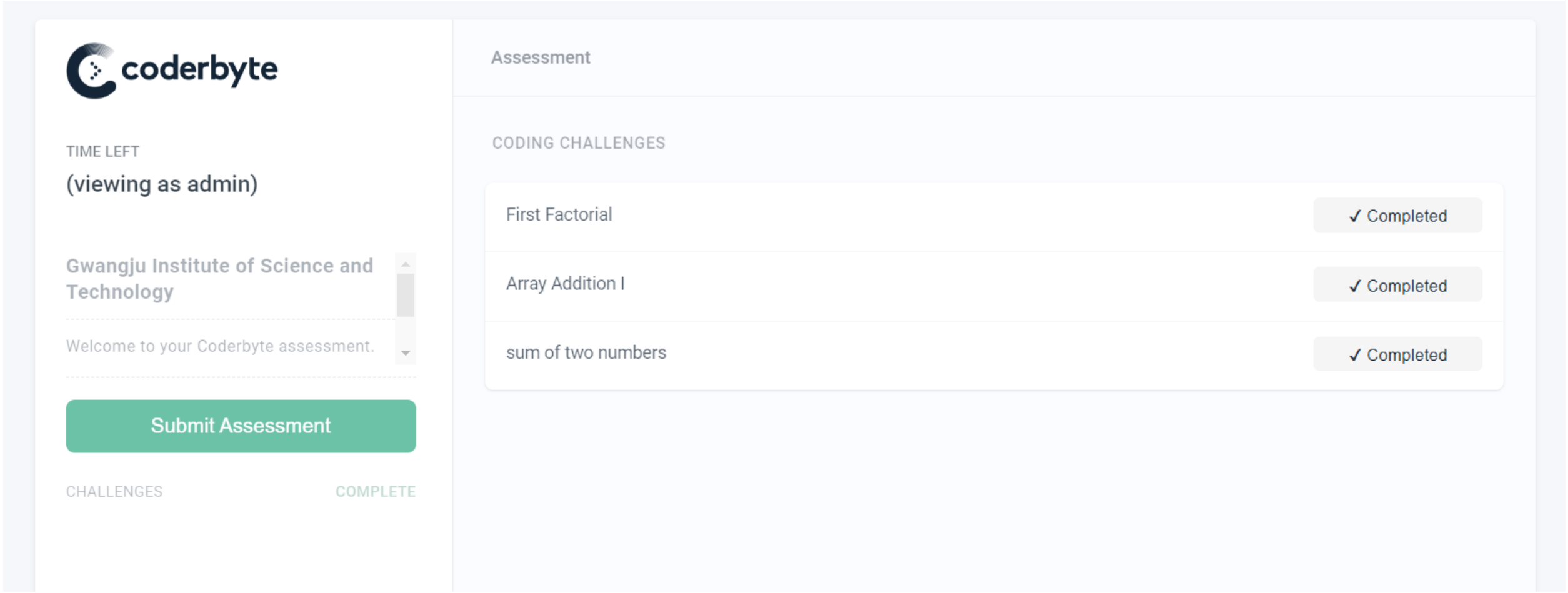
Write your code, test your code and submit



- Read and solve your challenge.
- You can run code, and provide input in the text box
- You can run all the test cases, at once, it will indicate if any test case is failing
- Fix your code and run it as many times as you like before clicking submit button

After completing all the challenge Submit Assignment

- After finishing all the assignment, you can click on Sumit Assignment



Professors Dashboard looks like this

Python Assessment

Preview Edit

Created by pawantiwari@bitmesra.ac.in on Jul 05 2023

Invite candidates (4/300 max)

john@email.com, sarah@email.com, ... (up to 50 email addresses)

Send private invite link OR Copy public invite link

Assessment details

3	0	0
Challenges	Open-ended	Multiple choice

Candidate details

25% Qualifying

4 invited
3 assessed
1 qualified

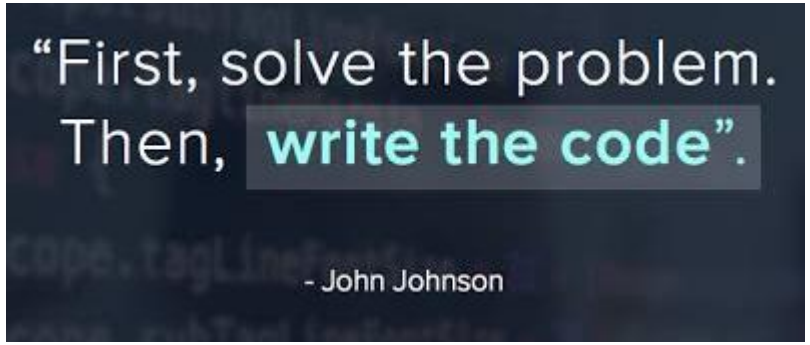
All candidates | JSON Export | CSV Export | Search for candidates...

Name	Email	Status	Joined	Time Taken	Final score	Cheating	Actions
20210234	suman17@gist.ac.kr	In progress	Aug 25	N/A	0%	Not detected	View report
suman	suman17july@gmail.com	Submitted	Jul 06	N/A	83%	Not detected	View report
-	pawantiwari@bitmesra.ac.in	Submitted	Jul 05	N/A	67%	Detected	View report
-	suman17@gist.ac.kr	Invited (Aug 25)	-	N/A	N/A	N/A	

Grading

- Assignment – 30%
- Mid Term – 30%
- Final Term – 30%
- Class presence – 10%

Teaching Methodology

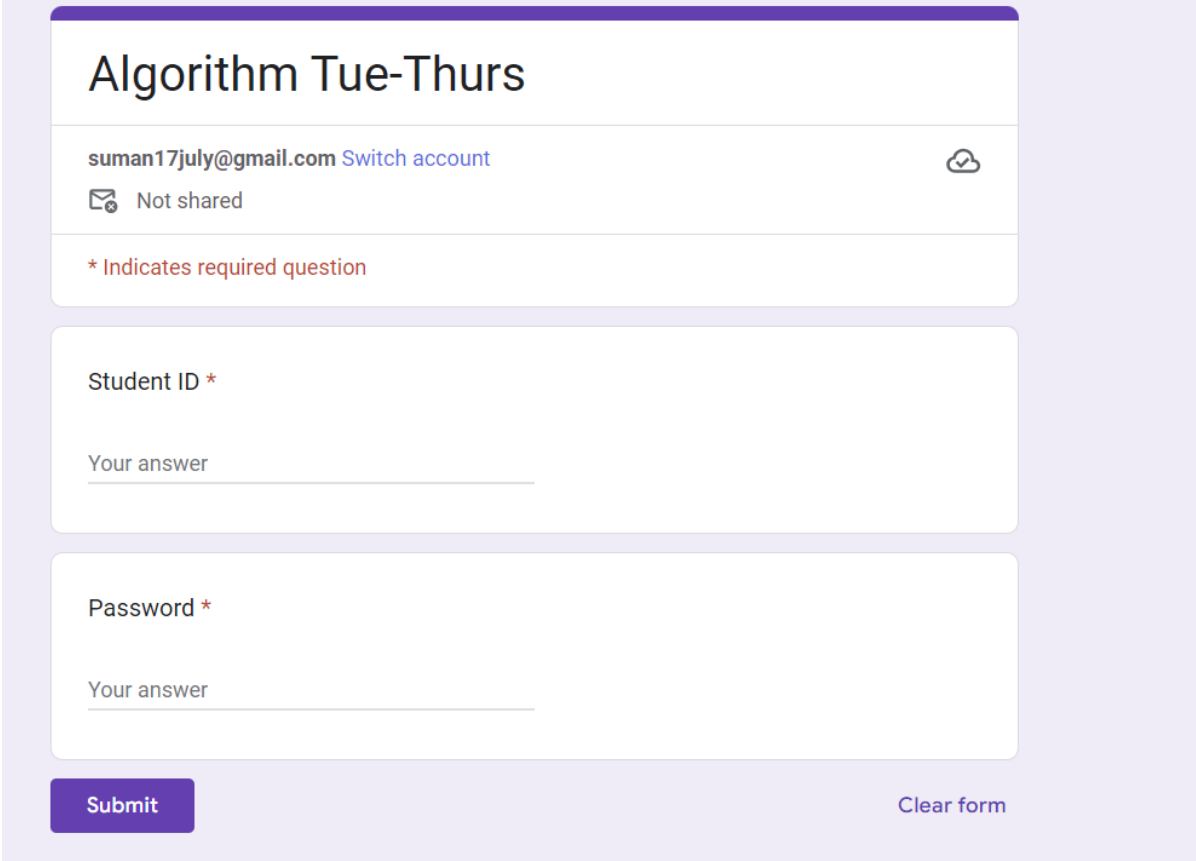


- I will first explain the algorithm, and then show you code
- There are two approach to this class
  - Mathematical
  - Coding
- Mathematical approach is great for writing paper and research articles
- Coding approach is better for preparing you for interview with big companies
- We will discuss both the approaches

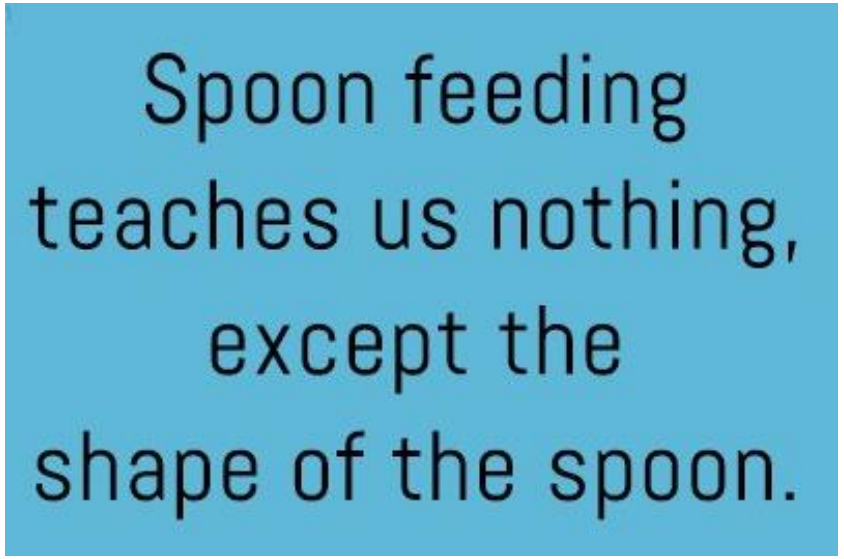
Attendance

Attendance will be taken in Google form. Please fill your Student ID carefully. Password for the atte ndance will be announced in the class.

- Algorithm Mon-Wed (google.com)
- Algorithm Tue-Thurs (google.com)



Expectation from Students



- **Code Practice**
  - **You should practice all the algorithm explained in the class by yourself, in order to do the assignment well**
  - **There wont be any Spoon Feeding.**
  - **Assignments will be re-evaluated by TA. Don't make their life hard. Do as expected.**
  - **Copying code is not allowed, the tool comes with plagiarism checker,**
    A very heave –ve penalty will be given to your if caught

What is Algorithm?

It's a set of computational steps, that transforms input to output.

Algorithm should have Definiteness, Finiteness, Effectiveness

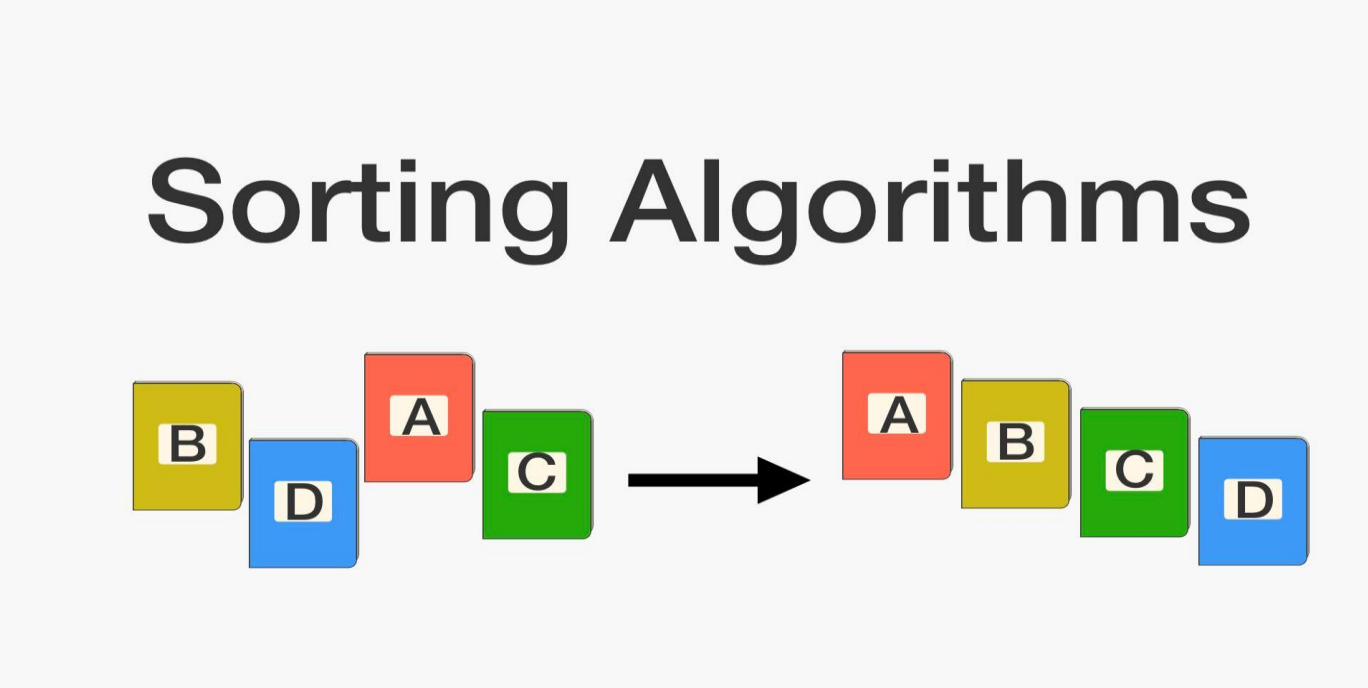
What kind of problems are solved using Algorithms

- **Human Genome Project**
  - Given two sequence of symbols, find longest common subsequence (Matching two DNA stands)
- **Internet**
  - Routing Algorithms
  - Search engines
- **Commerce**
  - Public-key cryptography
  - Digital signature generations
- **Maximizing profit**
  - Oil company- Where to put well to maximize profit
  - Election candidate – where to campaign to win election
  - Airlines – assign crew to flights in least expensive ways
  - ISP – where to install 5G devices to provide best quality to mobile users

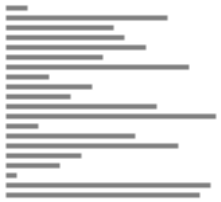
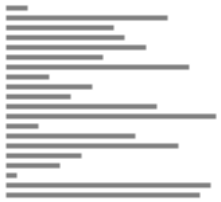
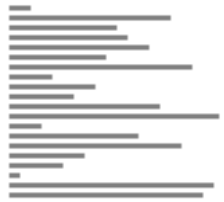
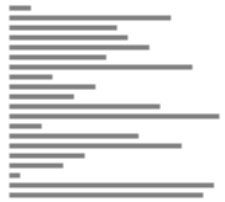
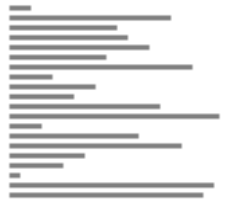
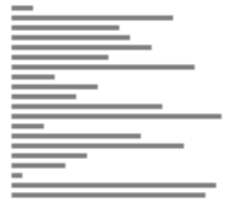
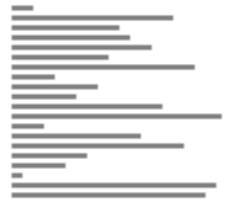
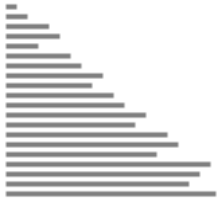
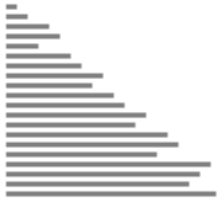
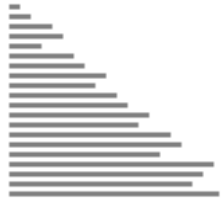
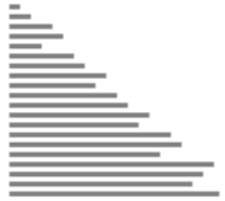
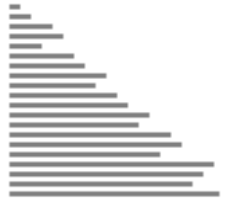
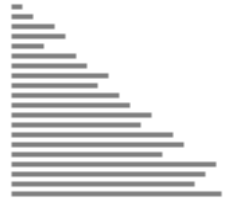
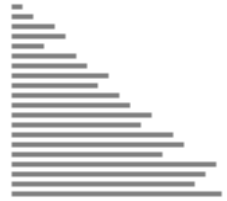
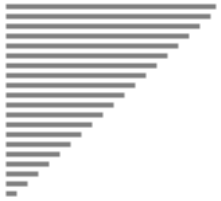
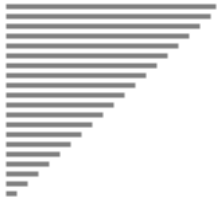
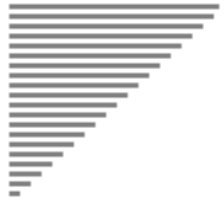
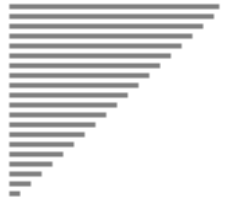
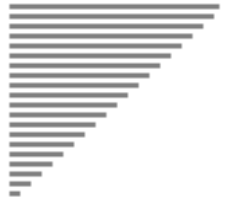
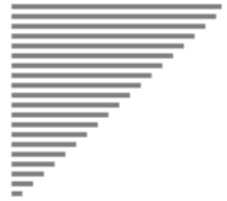
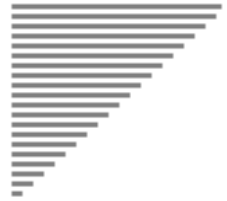
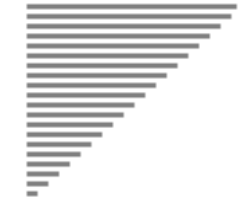
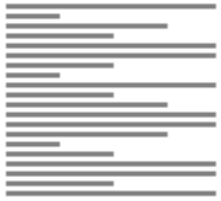
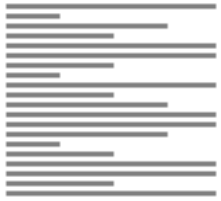
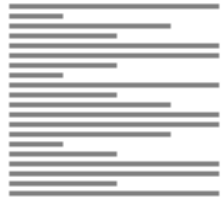
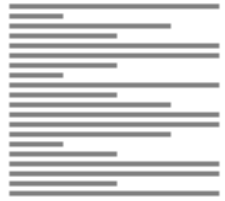
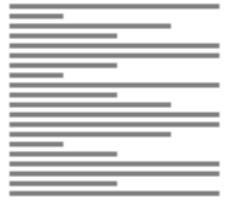
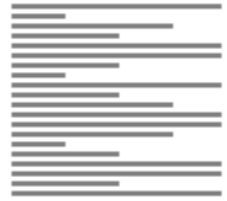
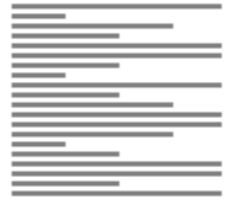
Difference between Data Structure and Algorithm

- **Data Structure**
  - It's a way to organize and store data.
  - No Single data structure works well for all purposes.
  - Its important to know the strength and limitations of several of them
  - Ex – Arrays, Lists, Tree, Graph, Stack, Queues, Hash tables, Maps
- **Algorithms**
  - Step by step procedure for performing a task.
  - Specific data structure can help design a good algorithm to solve the problem
  - Data structure is pre-requisite for Algorithm course

Lets Start



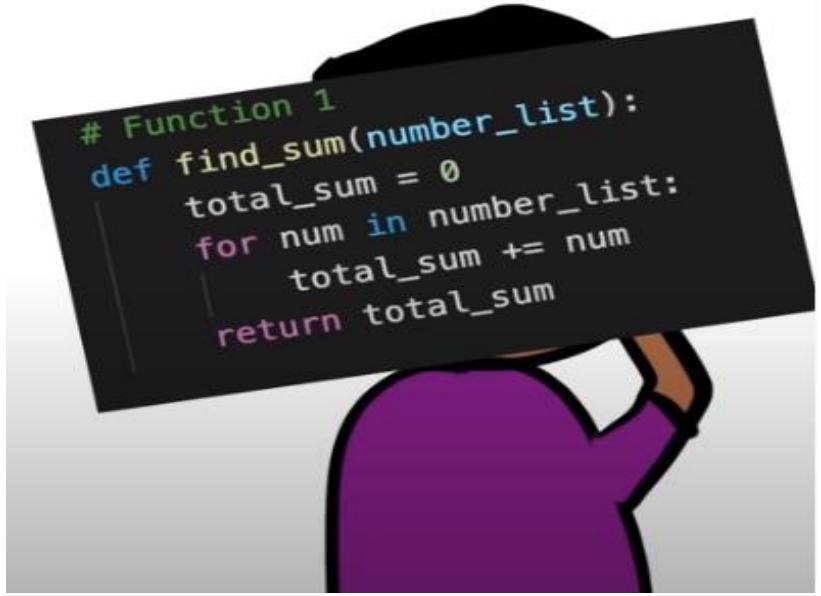
Types of Sorting Algorithm

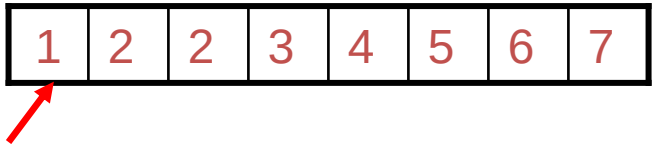
	Insertion	Selection	Bubble	Shell	Merge	Heap	Quick	Quick3
Random								
Nearly Sorted								
Reversed								
Few Unique								

- Criteria to test an algorithm - No of Operations. – Big 0 notation

Big 0

- Big 0 Notation
  - A mathematical notation used to classify algorithms according to how their **run time** or space requirements grow **as the input size grows**.





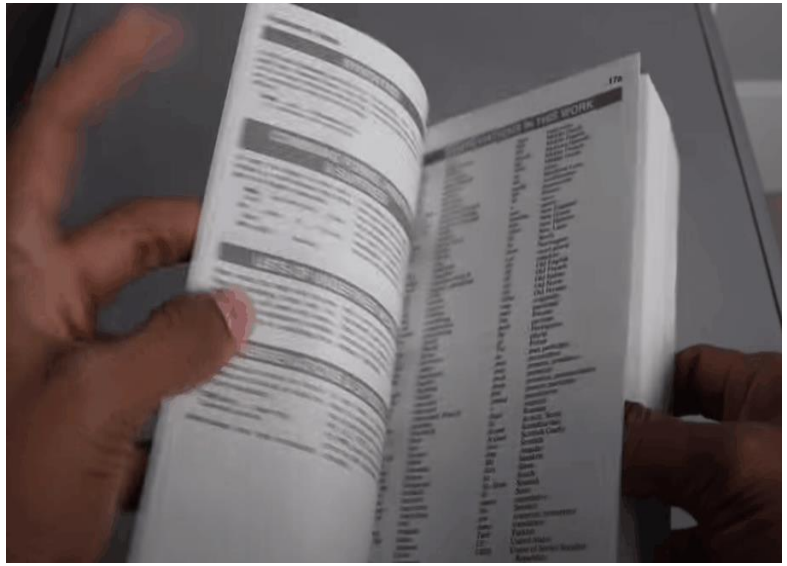
- 1 Operation - **O(1)**

n Operation - **O(n)**

Big 0 Searching Dictionary

- Big 0 Notation

Sequential Search

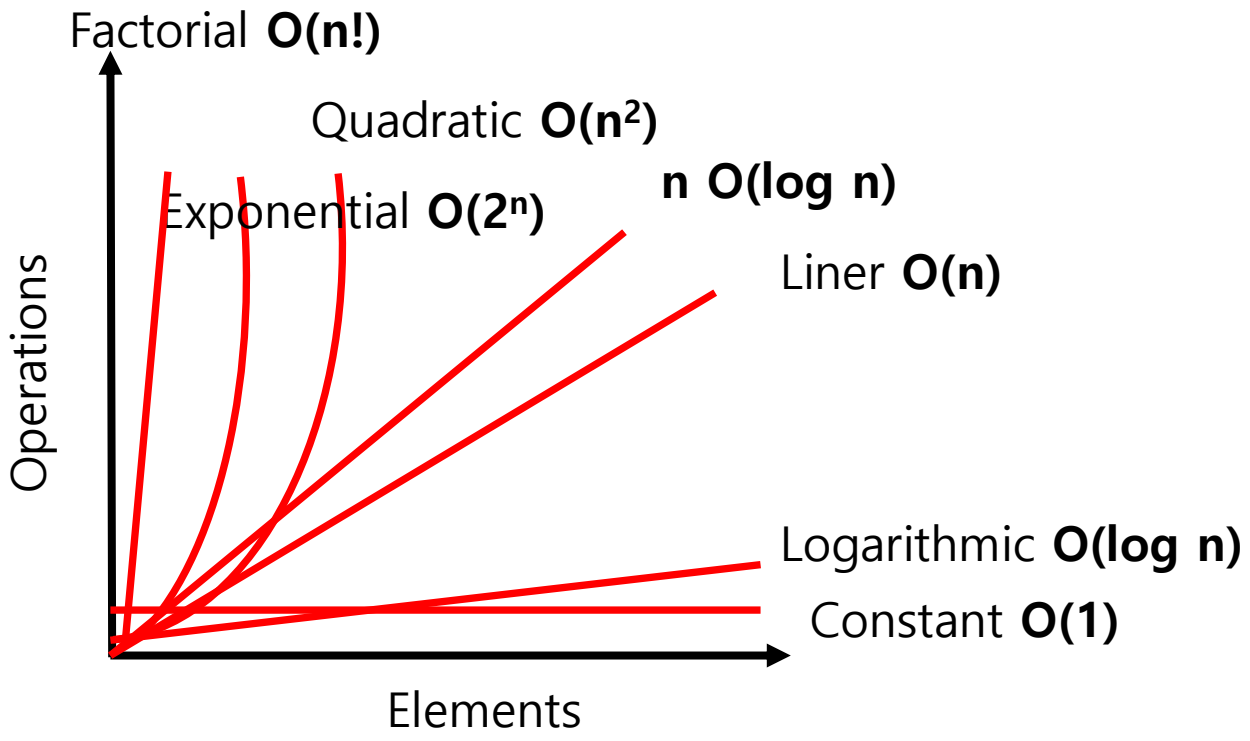


- 1 Operation - **O(n)**

Binary Search

n Operation - **O(log n)**

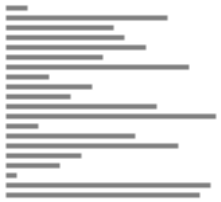
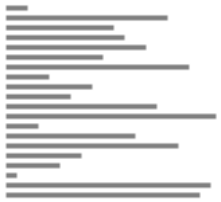
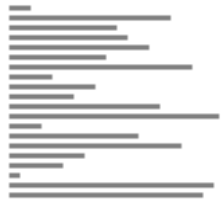
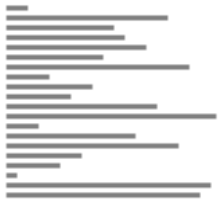
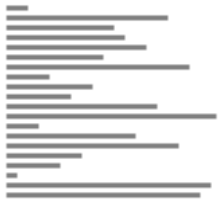
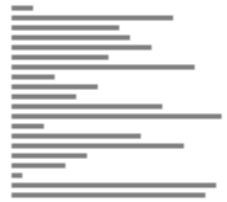
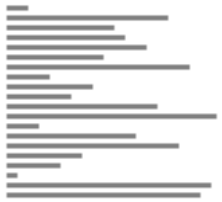
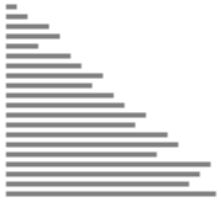
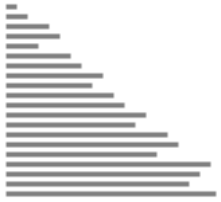
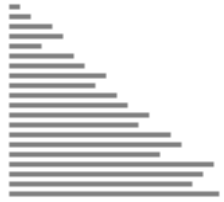
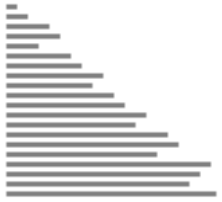
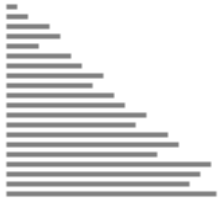
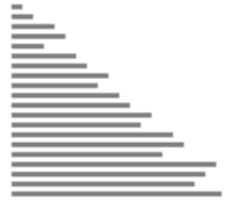
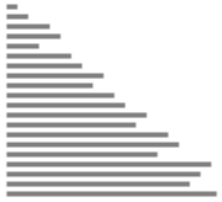
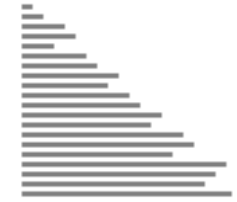
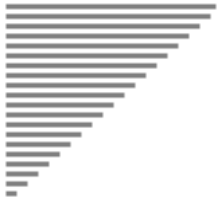
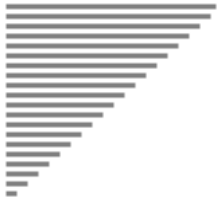
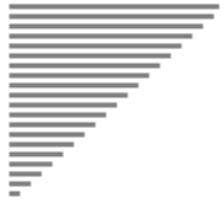
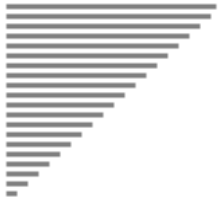
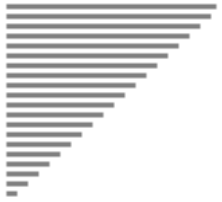
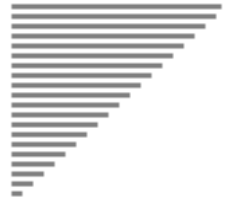
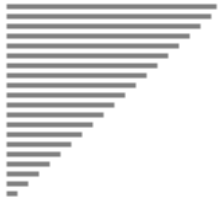
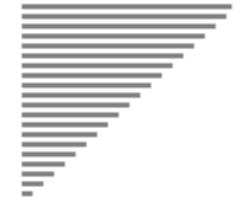
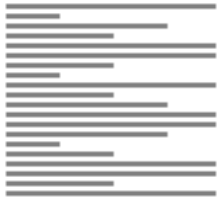
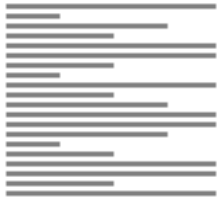
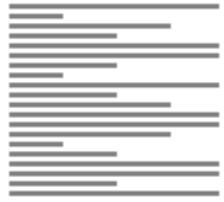
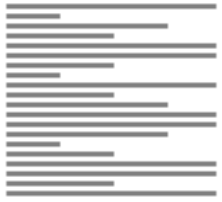
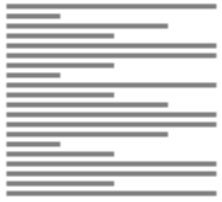
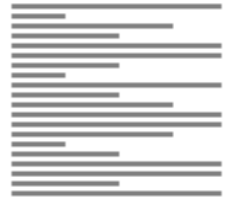
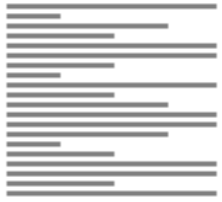
Time Complexities of Big 0



N: 17

O(1):	1
O(Log N):	4
O(N):	17
$O(N^2)$:	289
$O(2^N)$:	131072
O(N!):	355687428096(

Types of Sorting Algorithm

	Insertion	Selection	Bubble	Shell	Merge	Heap	Quick	Quick3
Random								
Nearly Sorted								
Reversed								
Few Unique								

- Criteria to test an algorithm - No of Operations. – Big 0 notation