بسم الله الرحمن الرحيم

BISMILLAH ARRAHMAN ARRAHEEM

CS302
Design and Analysis of Algorithms

Lecture 1: Complexity Analysis

Dr. Fahad Sherwani

(Assistant Professor – FAST NUCES)

fahad.sherwani@nu.edu.pk

About My Self

Education:

Doctorate of Philosophy (PhD) in Electrical Engineering, Universiti Tun Hussein Onn Malaysia (2019)
Specialization Cluster: Artificial Intelligence, Machine Learning, Algorithm Design
PhD Research Title: An Improved Data Classification Framework Based on Fractional Particle Swarm Optimization.

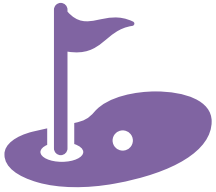
Master of Electrical Engineering, Universiti Tun Hussein Onn Malaysia (2013)
Specialization Cluster: Simultaneous Localization and Mapping (SLAM)
Master Research Title: Robotic Path Planning Using Rapidly-Exploring Random Trees.

Bachelor of Electronics Engineering, Hamdard University of Engineering and Technology Karachi Pakistan (2011)
Specialization: Embedded Systems & Robotics
Degree Research Title : Development of Tri-Wheeler Spy Robot.

Please Introduce Yourself

Your Good Name?

What are Algorithms?



Your Expectations from this course?

Course Objectives and Course Description

- Refer to PDF

Grading Scheme

- Midterm Exams (25%)
- Final exam (50%)
- Assignments (10%)
- Project (10%)
- Quizzes (05%)

Absolute Grading Scheme:

Total Marks (%)	Grade
≥ 90	A+
86-89	A
82-85	A-
78-81	B+
74-77	B
70-73	B-
66-69	C+
62-65	C
58-61	C-
54-57	D+
50-53	D
≤ 49	F

Teaching Material:

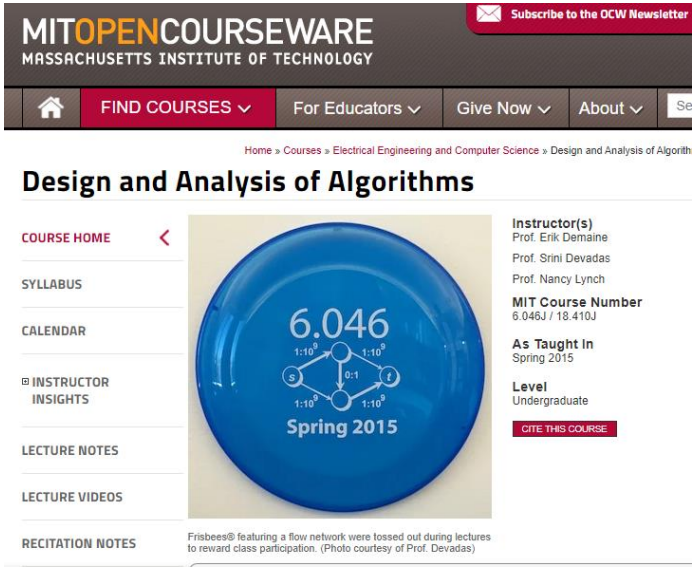
Textbooks:

Introduction to Algorithms, 3rd Edition

By Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein

Dasgupta, Papadimitriou, and Vazirani: **Algorithms**, McGraw-Hill, 2006

Additional Resources



https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-design-and-analysis-of-algorithms-spring-2015/index.htm

Introduction to Analysis of Algorithms

What is an Algorithm?

"Informally, an algorithm is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output".

An algorithm is thus a sequence of computational steps that transform the input into the output.

The algorithm describes a specific computational procedure
for achieving that input/output relationship.

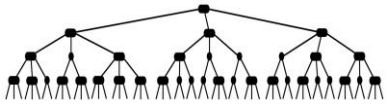
Introduction to Analysis of Algorithms

Why Analyze an Algorithm?

1. Classify problems and algorithms by difficulty.

2. Predict performance, compare algorithms, tune parameters.

3. Better understand and improve implementations and algorithms.



Intellectual challenge: AofA is even more interesting than programming!

Introduction to analysis of algorithms

Why Analyze an Algorithm?

•Analysis can be more reliable than experimentation. If we experiment, we only know the behaviour of a program on certain specific test cases, while analysis can give us guarantees about the performance on all inputs.

•It helps one choose among different solutions to problems. As we will see, there can be many different solutions to the same problem. A careful analysis and comparison can help us decide which one would be the best for our purpose, without requiring that all be implemented and tested.

•We can predict the performance of a program before we take the time to write code. In a large project, if we waited until after all the code was written to discover that something runs very slowly, it could be a major disaster, but if we do the analysis first we have time to discover speed problems and work around them.

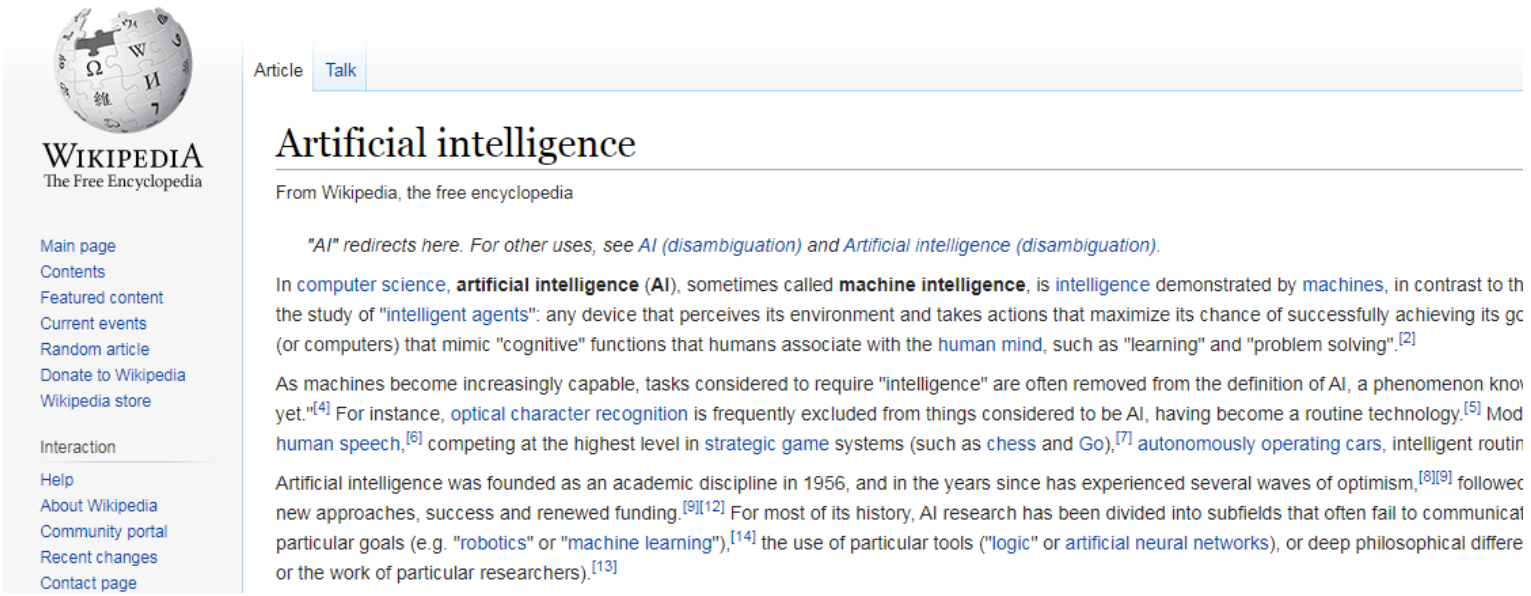
•By analysing an algorithm, we gain a better understanding of where the fast and slow parts are, and what to work on or work around in order to speed it up.

Applications of Algorithms

- Face Recognition
- Autonomous cars
- Social network analysis
- Recommendation systems
- Fraud detection
- Financial forecasting

…. **Many more!**

Read Article About Analysis of Algorithms



WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Article | Talk

Artificial intelligence

From Wikipedia, the free encyclopedia

"AI" redirects here. For other uses, see AI (disambiguation) and Artificial intelligence (disambiguation).

In computer science, **artificial intelligence** (**AI**), sometimes called **machine intelligence**, is intelligence demonstrated by machines, in contrast to th the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its gc (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving".[2]

As machines become increasingly capable, tasks considered to require "intelligence" are often removed from the definition of AI, a phenomenon kno yet."[4] For instance, optical character recognition is frequently excluded from things considered to be AI, having become a routine technology.[5] Mod human speech,[6] competing at the highest level in strategic game systems (such as chess and Go),[7] autonomously operating cars, intelligent routin

Artificial intelligence was founded as an academic discipline in 1956, and in the years since has experienced several waves of optimism,[8][9] followec new approaches, success and renewed funding.[9][12] For most of its history, AI research has been divided into subfields that often fail to communicat particular goals (e.g. "robotics" or "machine learning"),[14] the use of particular tools ("logic" or artificial neural networks), or deep philosophical differe or the work of particular researchers).[13]

https://en.wikipedia.org/wiki/Analysis_of_algorithms

Assignment 1

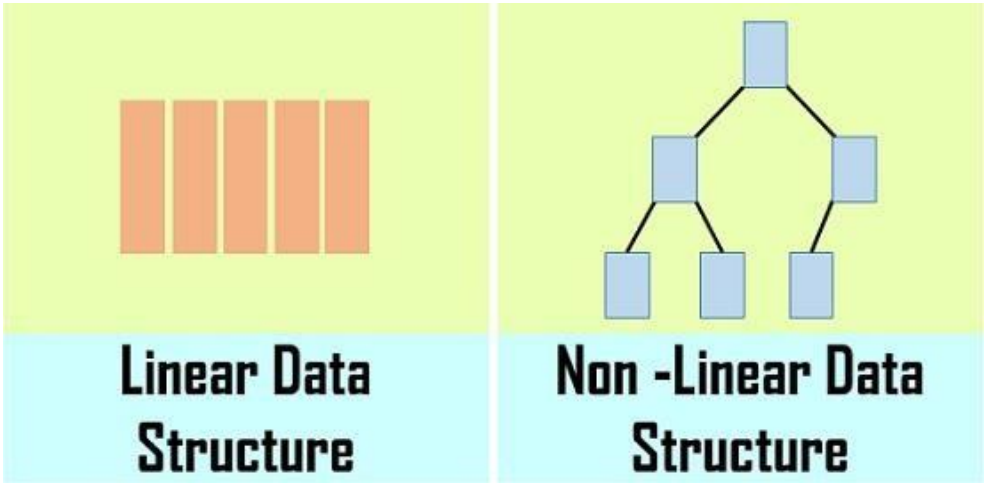
- What is the common definition of “Algorithms”? Do you agree?
- Do you know any application of Algorithms?
- Should artificial intelligence simulate natural intelligence?
- What are the criticisms on the AI research? Do you agree?
- What is the relation between AI and logic? AI and philosophy? Logic and philosophy?
- Explain the meaning of logic? reasoning? ontology?
- What is Natural Language Processing? And how it is related to AI?
- Why and how Probabilistic and statistical methods are used in AI ?
- What are the major research approaches/schools in AI? Which one you think is more productive?

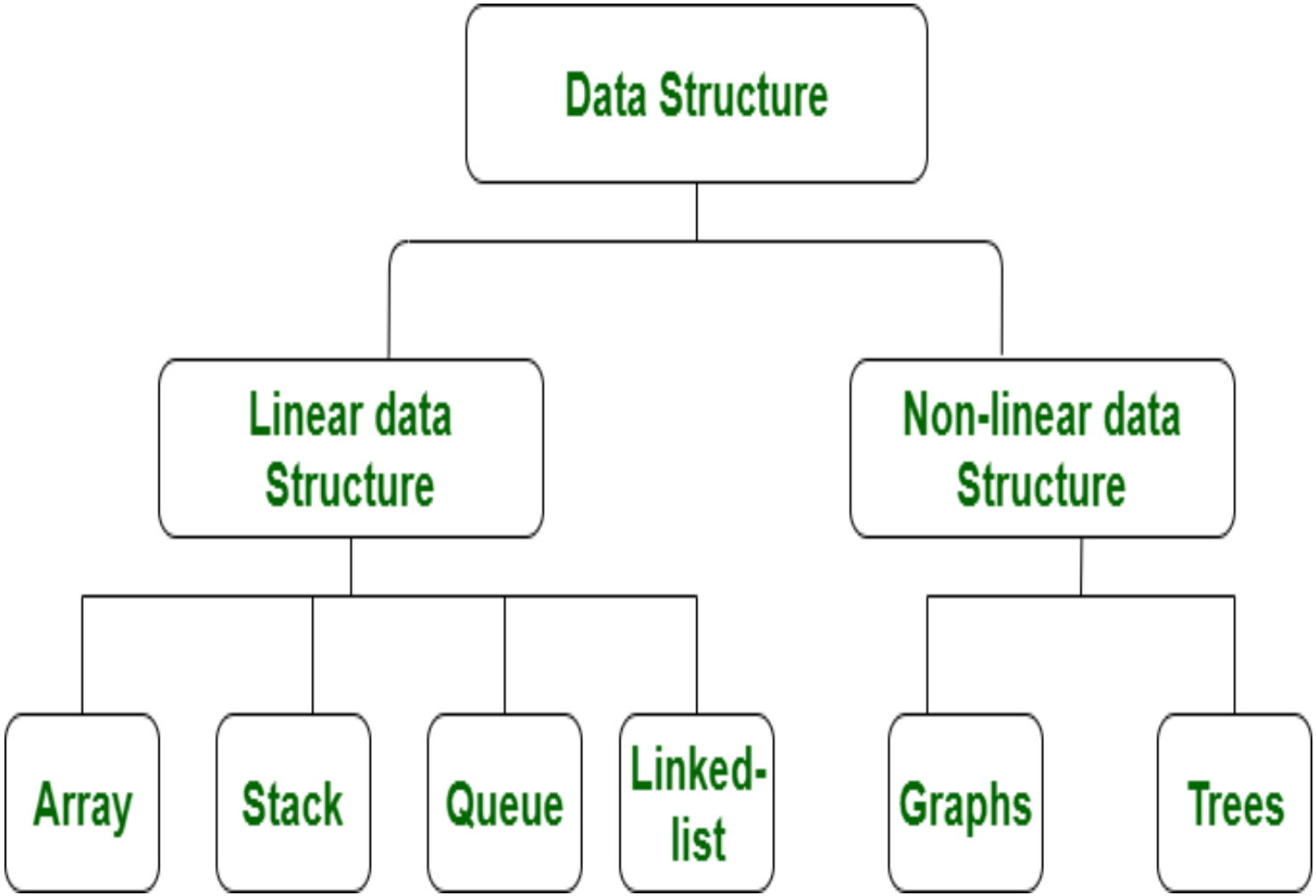
What are data structures?

- A way of organizing data that is stored in computer.
- Several types of data structures are available.
- Each data structure has:
  - Specific organization of data
  - Specific ways of accessing /manipulating data
  - Specific amount of memory allocated

Types of Data Structure

1. Linear Data Structures: A linear data structure organizes data in sequential manner i.e. with respect to indexes /indices
2. Non-Linear Data Structures: A non-linear data structure organizes data in hierarchical manner.




Data Structure
Linear data Structure
Non-linear data Structure
Array
Stack
Queue
Linked-list
Graphs
Trees

What are algorithms?

- An algorithm is a finite sequence of steps for solving a particular problem.
- Two major properties of an algorithm are:
  - Definiteness: Well defined steps
  - Finiteness: It should complete in finite amount of time. It should not enter in any infinite loop

Why the knowledge of Data Structure is so important?

- **Data structures** provide a means to manage large amounts of **data** efficiently
- Examples include several national and international databases , data utilised by social networking platforms and other business .
- Usually, efficient **data structures** are key to designing efficient algorithms

Why the knowledge ofData Structure is so important?

- Each program depends on data. Not every type of data can be stored in variables. More sophisticated structures are required to hold data.
- If a data is stored in inefficient data structure then program will never be able to utilise the multiprocessor systems.

Why the knowledge ofData Structure is so important?

- Search for value of 896 or 273
  - If the data is sorted
  - If data structure facilitates Random access
  - If the data is not sorted
  - If data structure facilitates sequential access

456	896	273	888	746

Types of operations

- Searching
- Insertions
- Deletions
- Sorting
- Splitting
- Merging
- Accessing

Basic Data Structures

- Array
- Stack
- Queue
- Tree
- Graph
- Hash Table

Algorithms and It's Complexity Analysis

- A good algorithm consumes less resources
- Resources include:
  - Running Time
  - Memory
  - Other Hardware
  - Communication time

  In this course, our major concern is the running time.

Running Time of an Algorithm

- Running time depends on
  - Input to the algorithm (size and nature of input)
  - Hardware Environment (Processor ,clock rate, memory ,disk etc)
  - Software Environment in which algorithm is implemented, compiled and executed (operating system, programming language, compiler and interpreter)

Ways of Measuring the Running Time of Algorithm

1. Empirical or Experimental Analysis:

In this technique, running time of algorithm is measured by executing it on various test inputs and recording the actual time spent in each execution.

The limitations of this approach are:

- Difficulty to compare the efficiency of two algorithms unless experiments are performed in the same environment with respect to hardware and software.

Limitations of Empirical Approach

- It is necessary to implement and execute an algorithm to study its running time experimentally on all possible combinations of hardware environment, software environment and test input types which is not possible.

Ways of Measuring the Running Time of Algorithm

1. Empirical Analysis
2. Theoretical Analysis:
   - This approach uses high level description of algorithm instead of implementing that algorithm.
   - Characterizes running time as function of input size
   - Takes into account all possible inputs
   - Speed of an algorithm can be calculated independent of both hardware and software environment.

Primitive Operations

- Primitive operations are basic or elementary operations for example:
  - Adding two numbers
  - Comparing two numbers
  - Assigning a value to variable
  - Calling a method or function
  - Returning from a method or function

Counting number of primitive operations is another approach of calculating complexity of algorithm

Calculating Complexity of Algorithm

Finding Sum of all elements of an array:

Algorithm : FindSum (A, n)
Input : An array 'A' of 'n' integers
Output : Sum of all integers in 'A'

STEPS

1- Sum = A[0]
2- for i = 1 to i = n-1
3- Sum = Sum + A[i]
4- return Sum

step 2 can also be written as :-

for ($\underbrace{i = 1}_{2a}$; $\underbrace{i <= n-1}_{2b}$, $\underbrace{i++}_{2c}$)

Calculating Complexity of Algorithm

Statement#	Operations	Iterations	Subtotal
1	2	1	$2 \times 1 = 2$
2a	1	1	$1 \times 1 = 1$
2b	1	n	$1 \times n = n$
2c	2	$n-1$	$2 \times (n-1) = 2n-2$
3	3	$n-1$	$3 \times (n-1) = 3n-3$
4	1	1	$1 \times 1 = 1$

$$\text{Complexity} = 2 + 1 + n + 2n - 2 + 3n - 3 + 1$$
$$= 6n - 1$$

Best /Worst /Average cases

Based on the input data , performance of algorithm can be categorised into three classes:

1. Best Case:

This happens when the input is such that the algorithm runs in the shortest possible time.

Best /Worst /Average cases

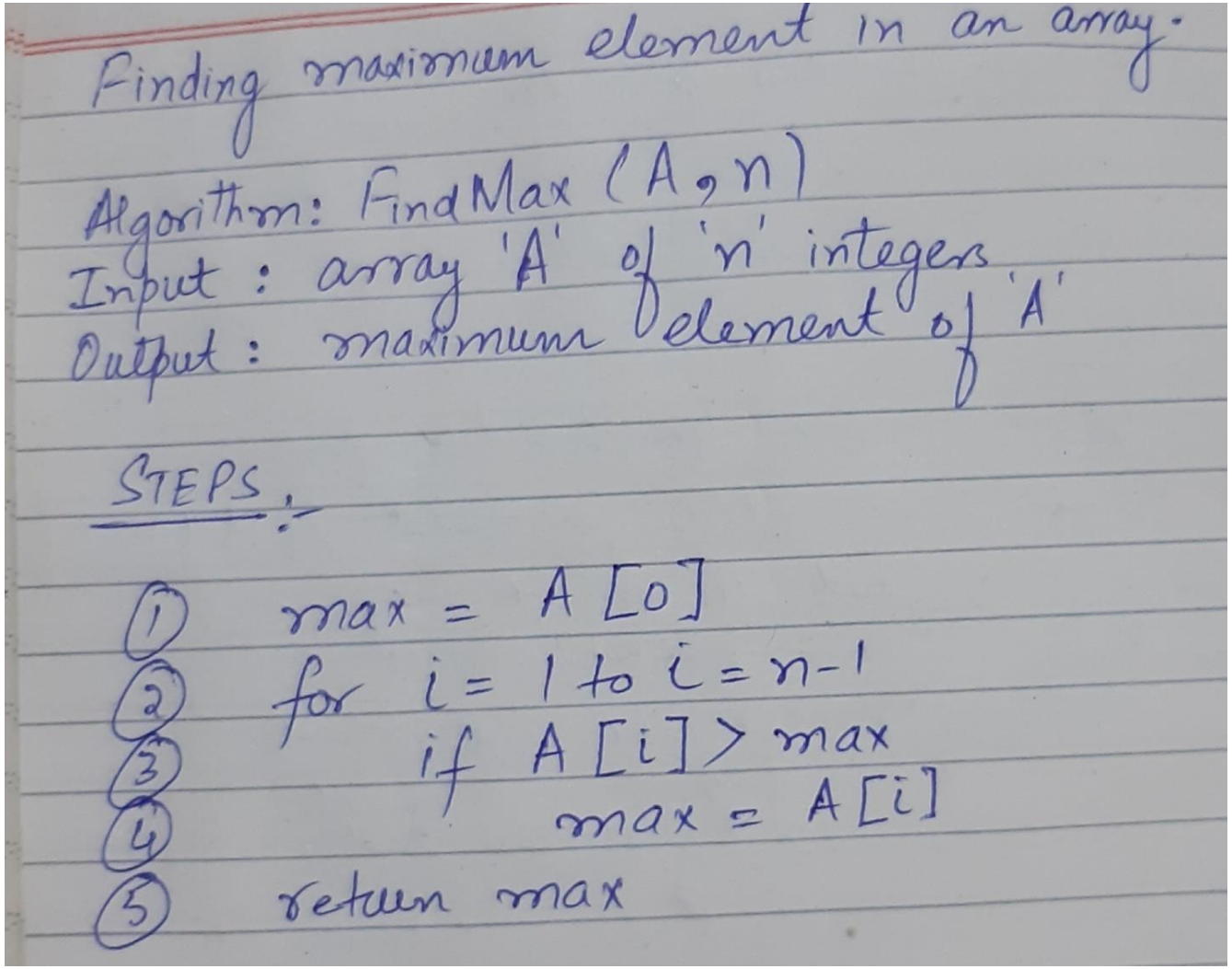
2. Worst Case

This happens when the input is such that the algorithm runs in the longest possible time.

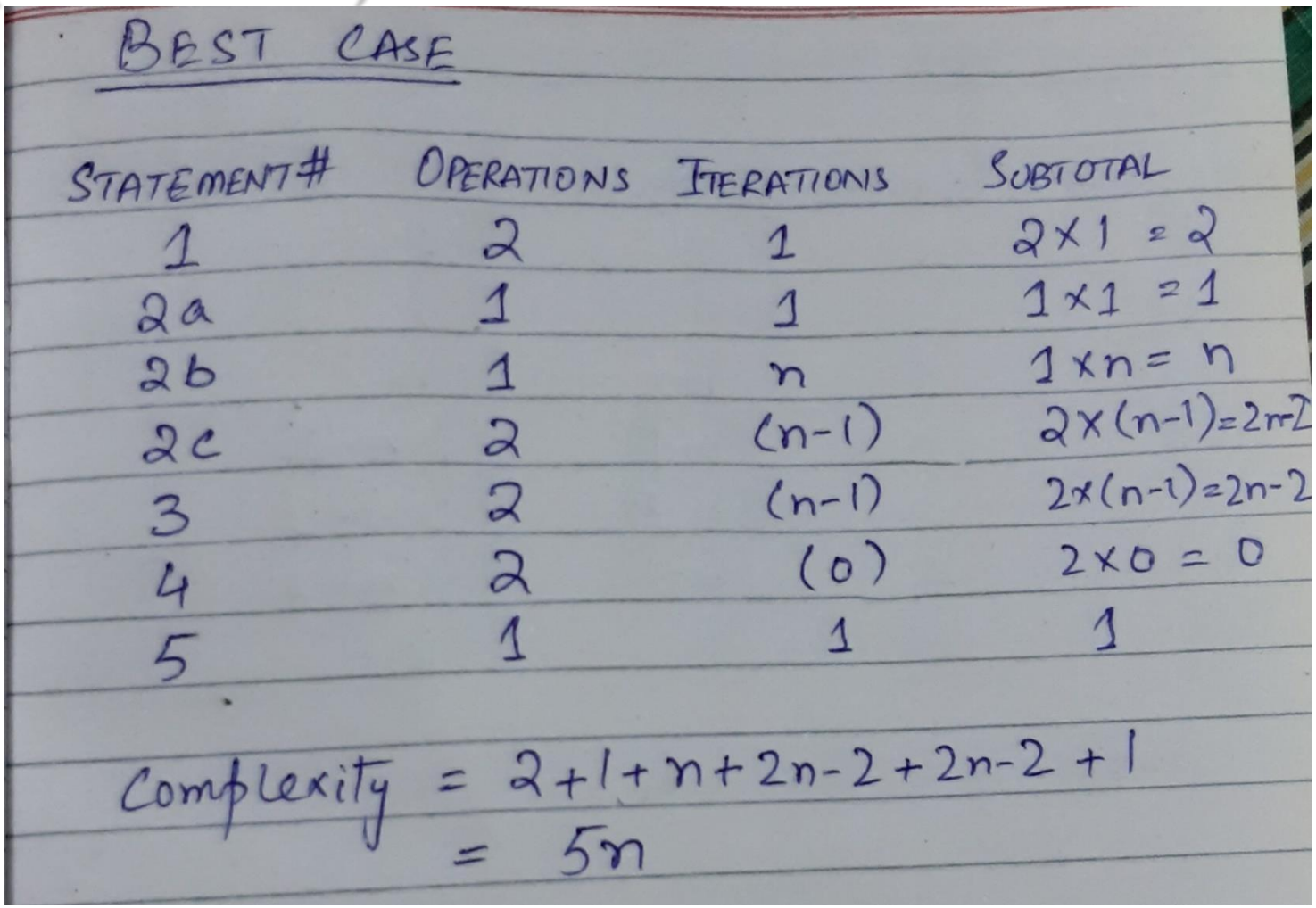
3. Average Case:

An algorithm may run faster on some inputs than it does on others. In such cases, we express the running time of such an algorithm as an average taken over all possible inputs.

Calculating Complexity of Algorithm



Calculating Complexity of Algorithm (Best case)



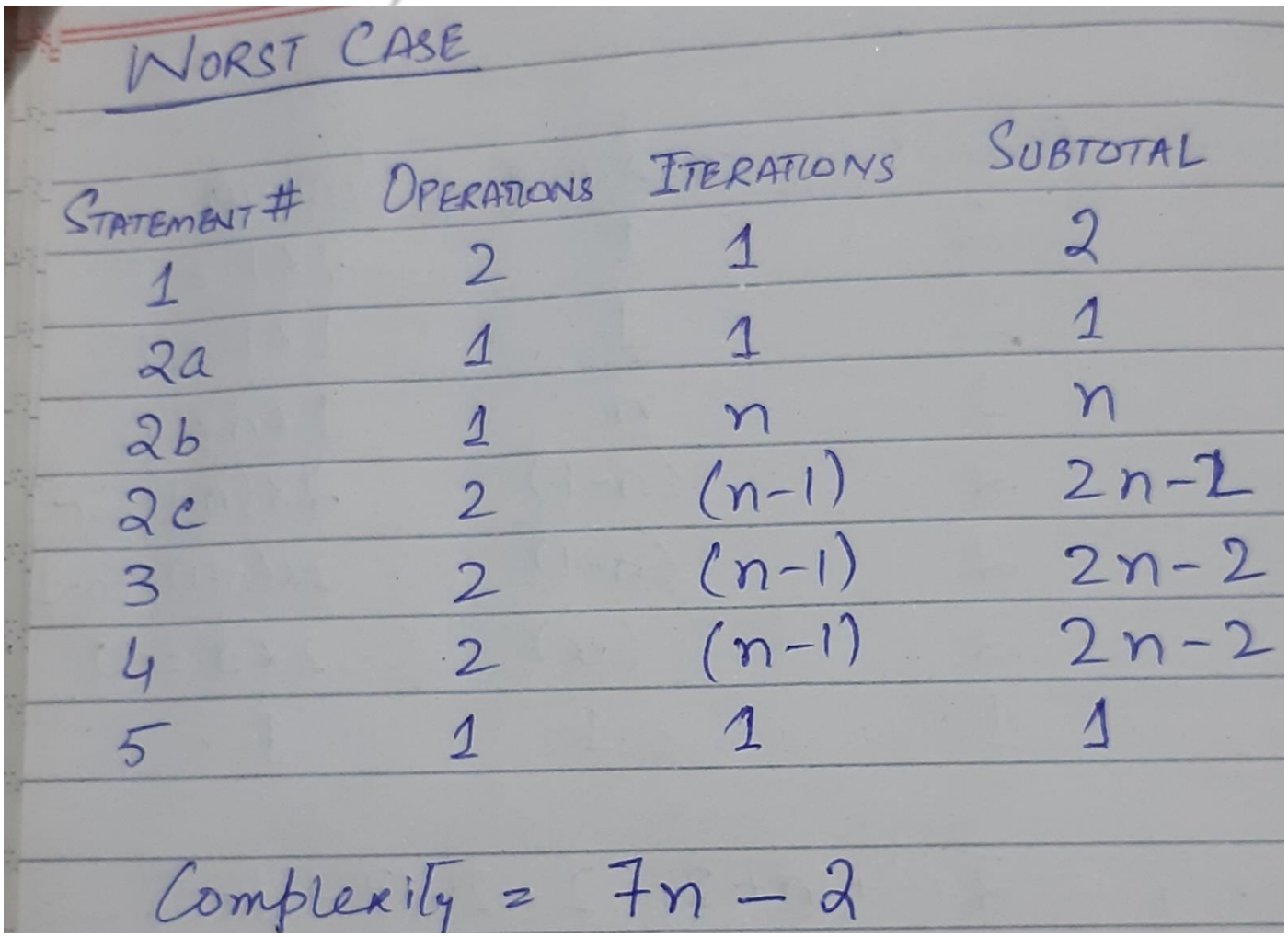
BEST CASE

STATEMENT#	OPERATIONS	ITERATIONS	SUBTOTAL
1	2	1	2×1 = 2
2a	1	1	1×1 = 1
2b	1	n	1×n = n
2c	2	(n-1)	2×(n-1)=2n-2
3	2	(n-1)	2×(n-1)=2n-2
4	2	(0)	2×0 = 0
5	1	1	1

Complexity = 2+1+n+2n-2+2n-2+1

= 5n

Calculating Complexity of Algorithm (worst case)

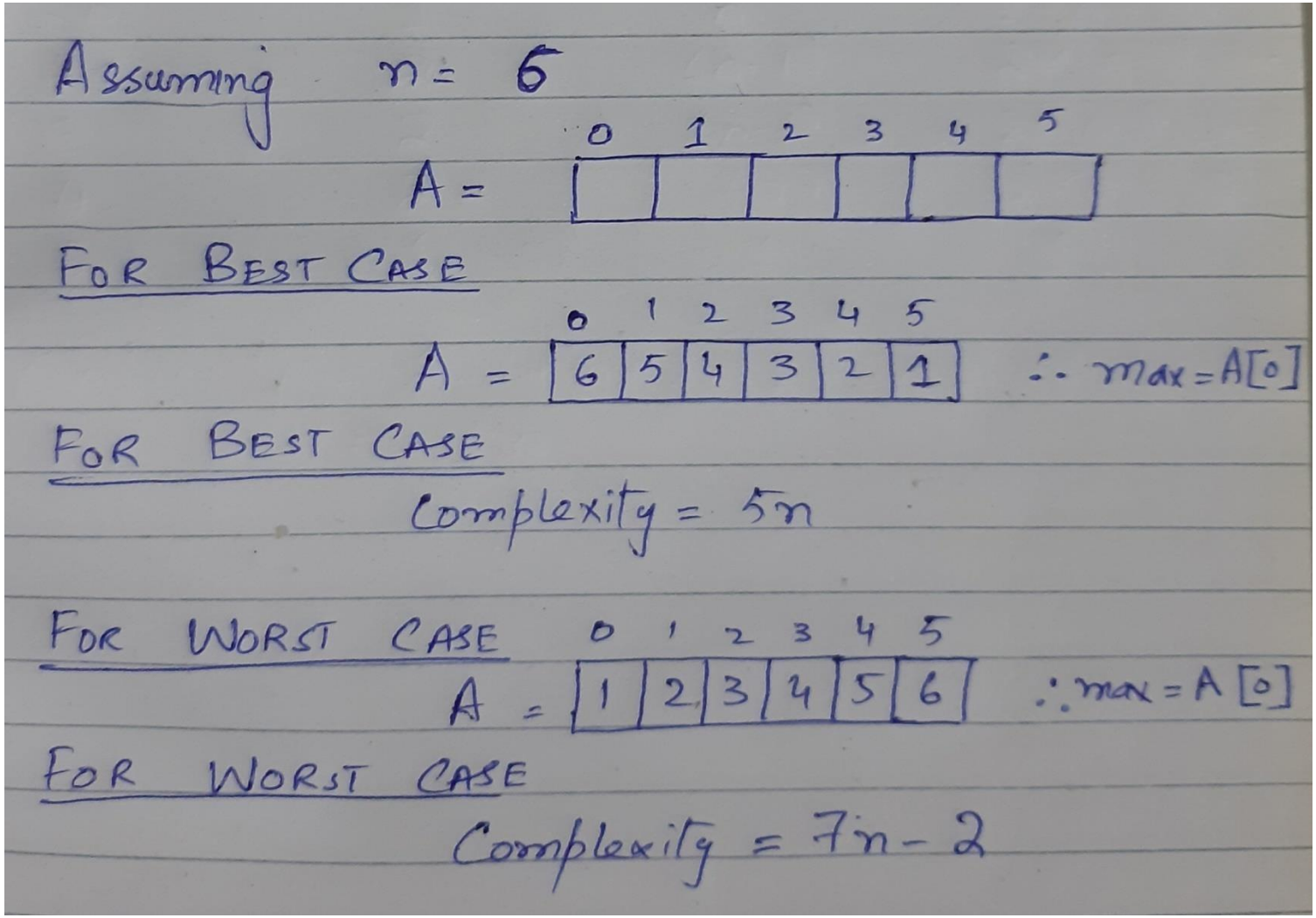


WORST CASE

STATEMENT #	OPERATIONS	ITERATIONS	SUBTOTAL
1	2	1	2
2a	1	1	1
2b	1	n	n
2c	2	(n-1)	2n-2
3	2	(n-1)	2n-2
4	2	(n-1)	2n-2
5	1	1	1

Complexity = $7n - 2$

Calculating Complexity of Algorithm



After today's lecture.....

- You should
  - Know the linking of data structure with algorithms
  - Significance of studying data structure and algorithms
  - Know the differentiate between Empirical complexity analysis and theoretical complexity analysis
  - Know how to calculate the complexity in terms of best, worst and average cases.