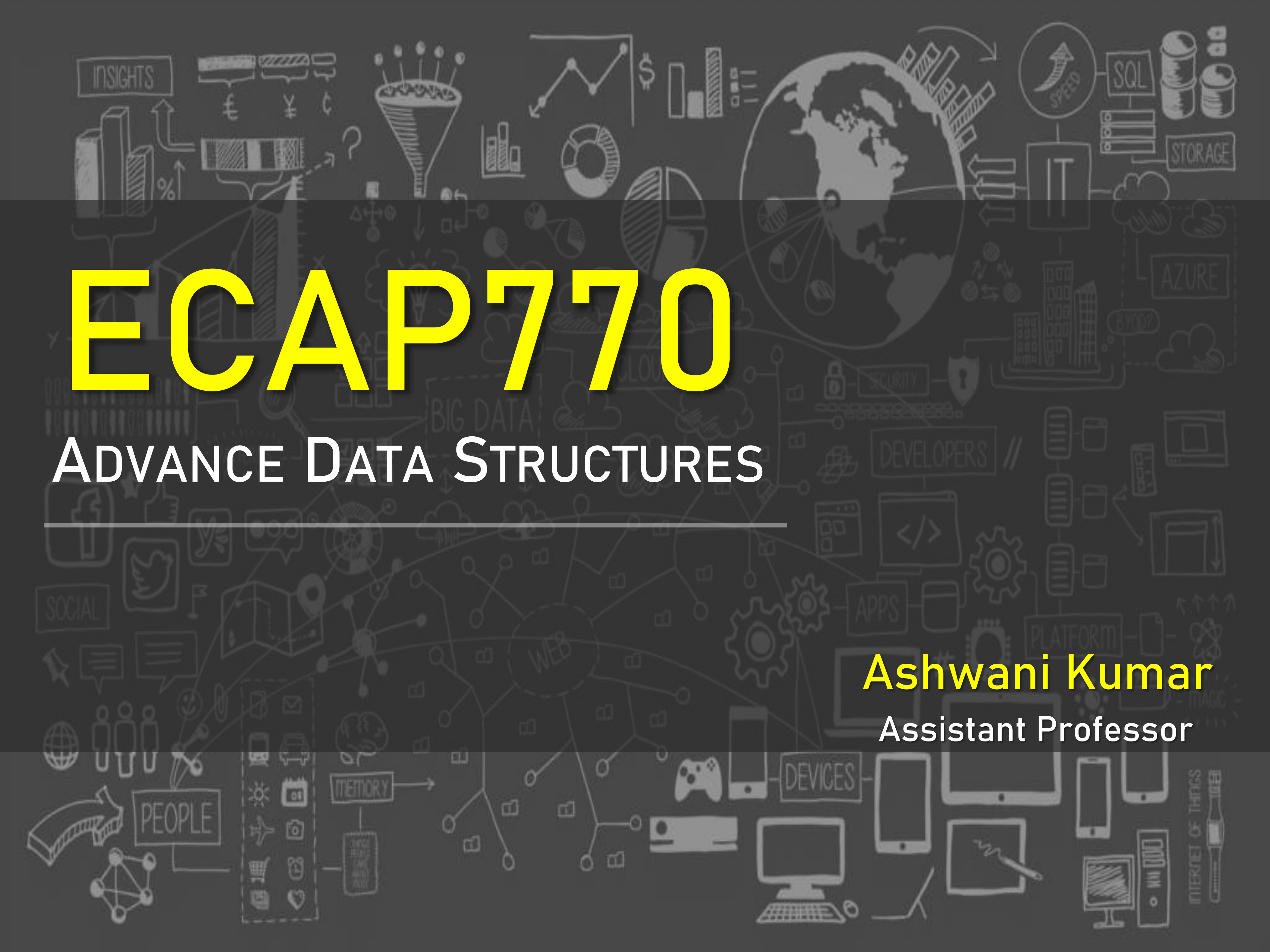

ECAP770

Advance Data Structures

Ashwani Kumar

Assistant Professor

Learning Outcomes

After this lecture, you will be able to

- know about Algorithm

Algorithm

- Algorithm is set of rules/ instructions that step-by-step define how a work is to be executed upon in order to get the expected results.
- systematic procedure that produces in a finite number of steps the answer to a question or the solution of a problem.

Algorithm

- Computer algorithms work via input and output. They take the input and apply each step of the algorithm to that information to generate an output.
- E.g. a search engine is an algorithm that takes a search query as an input and searches its database for items relevant to the words in the query. It then outputs the results.

Characteristics of an Algorithm

- Well defined Input and output
- Clear and Unambiguous
- Finite-ness
- Feasible
- Language Independent

Characteristics of an Algorithm

- Input and output should be defined precisely.
- Each step in the algorithm should be clear and unambiguous.
- Algorithms should be most effective among many different ways to solve a problem.
- An algorithm shouldn't include computer code. Instead, the algorithm should be written in such a way that it can be used in different programming languages.

Characteristics of an Algorithm

- The algorithm must be finite, i.e. it should not end up in an infinite loops or similar.
- The algorithm must be simple, generic and practical, such that it can be executed upon will the available resources. It must not contain some future technology, or anything.
- The Algorithm designed must be language-independent, i.e. it must be just plain instructions that can be implemented in any language, and yet the output will be same, as expected.

Types of Algorithms

Algorithms are categorized based on the concepts that they use to accomplish a task.

- Divide and conquer algorithms
- Brute force algorithms
- Greedy algorithms
- Backtracking algorithms
- Randomized algorithms

Algorithm: Example

- Step 1: Start
- Step 2: Declare variables num1, num2 and sum.
- Step 3: Read values num1 and num2.

Algorithm: Example

- Step 4: Add num1 and num2 and assign the result to sum.
    - Sum = num1+num2
- Step 5: Display sum
- Step 6: Stop

Algorithm Complexity

Space Complexity

Time Complexity

Space Complexity

- Space Complexity: Space complexity of an algorithm refers to the amount of memory that this algorithm requires to execute and get the result. This can be for inputs, temporary operations, or outputs.
- Fixed Part: This refers to the space that is definitely required by the algorithm. For example, input variables, output variables, program size, etc.

Space Complexity

- Variable Part: This refers to the space that can be different based on the implementation of the algorithm. For example, temporary variables, dynamic memory allocation, recursion stack space, etc.

Time Complexity

- Time Complexity: Time complexity of an algorithm refers to the amount of time that this algorithm requires to execute and get the result. This can be for normal operations, conditional if-else statements, loop statements, etc.
- Constant time part: Any instruction that is executed just once comes in this part. For example, input, output, if-else, switch, etc.
- Variable Time Part: Any instruction that is executed more than once, say n times, comes in this part. For example, loops, recursion, etc.

Advantages of Algorithms

- It is easy to understand and help to solve problem.
- Algorithm is a step-wise representation of a solution to a given problem.
- In Algorithm the problem is broken down into smaller pieces or steps hence, it is easier for the programmer to convert it into an actual program.

Disadvantages of Algorithms

- Writing an algorithm takes a long time so it is time-consuming.
- Branching and Looping statements are difficult to show in Algorithms.

That’s all for now…