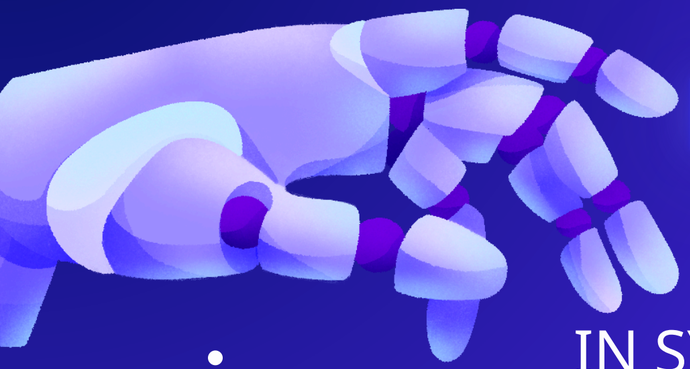

ARTIFICIAL INTELLIGENCE
BY : LAILA KAMAL ELDIN

WHAT IS COMPLEXITY??

The term "complexity" refers to a scientific theory that examines how certain systems display behaviors that are difficult to explain through traditional analysis of their individual parts. These behaviors, known as emergent properties, arise from the interactions within the system and cannot be predicted simply by understanding the system's components. Examples of complex systems include ecosystems, the human brain, financial markets, and social networks



HOW TO CREATING COMPLEXITY??

- IN SYSTEMS AND MODELS

  - Understanding Complexity

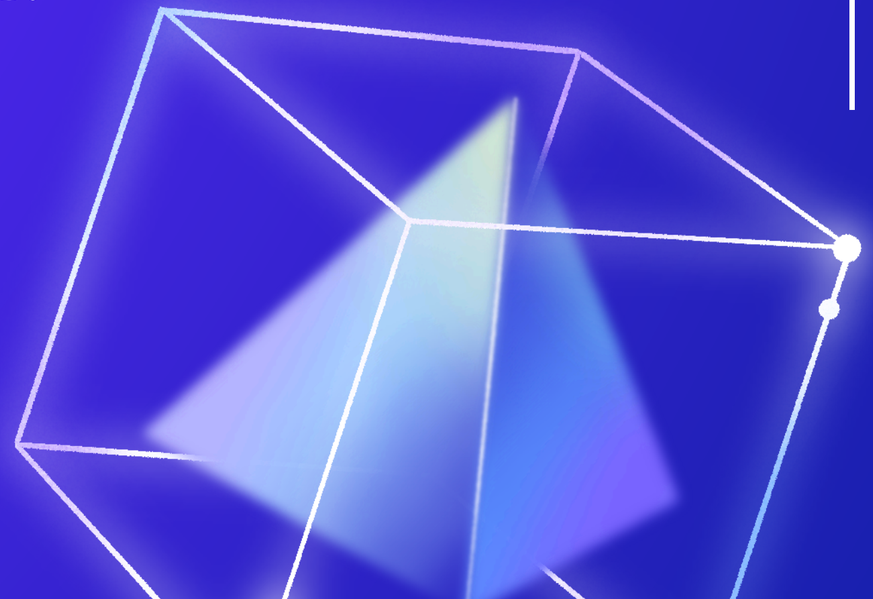
INTERCONNECTEDNESS: CREATE A WEB OF DEPENDENCIES BETWEEN ELEMENTS.

NON-LINEARITY: INTRODUCE RELATIONSHIPS WHERE SMALL CHANGES LEAD TO DISPROPORTIONATE EFFECTS.

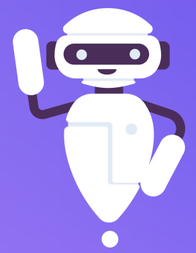
FEEDBACK LOOPS: ALLOW OUTPUTS TO INFLUENCE INPUTS, CREATING CYCLICAL PATTERNS.

EMERGENCE: LET COMPLEX PATTERNS ARISE FROM SIMPLE INTERACTIONS.

RANDOMNESS: INCORPORATE UNPREDICTABLE ELEMENTS.



PROJECT OBJECTIVES



TIME COMPLEXITY: O(N^2)

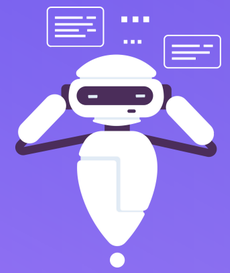
Nested Loop:

```
for i in range(n):
for j in range(n)
```

TIME COMPLEXITY: O(N^2)

```
matrix = [
  [1, 2, 3],
  [4, 5, 6],]

for i in range(len(matrix)):
 for j in range(len(matrix[i])):
    print(matrix[i][j])
    print()
```



TIME COMPLEXITY: O(LOG N)

```
def divide conquer(n):
if n <= 1:
return
divide conquer(n / 2)
```

COMPLEXITY OF BINARY SEARCH

01

- Best Case: O(1)

The best case occurs when the target value is the middle element of the array in the first comparison.

02

- Worst Case: O(log n)

The worst case occurs when the algorithm must search through all levels of the binary search tree. This happens when the element is not in the array or is found after many divisions.

TYPES OF SEARCH

01

Hash Table Search:
Description: Uses a hash function to index into an array for quick lookup.

- Best Case: O(1).
- Worst Case: O(n)

02

Depth-First Search (DFS)
Description: Explores as far as possible along each branch before backtracking.

- Best Case: O(1)
- Worst Case: O(V + E)

03

Breadth-First Search (BFS)
Description: Explores all neighbor nodes at the present depth level before moving on to nodes at the next depth level.

- Best Case: O(1)
- Worst Case: O(V + E)

04

Exponential Search:
Description: Finds the range where the target might be using exponential growth, then performs binary search within that range.

- Best Case: O(1)
- Worst Case: O(log n)