**DAY 6:**

**RECURSION:**

* Calling a method itself until a base case is reached.
* Recursion is used to solve breaking down into smaller, smaller sub-problems.
* Base case: stop the call otherwise it will continue.
* Reduce code length and complexity compared to iteration.
* Example: GCD, factorial, sum of digits

**BACKTRACKING:**

* It is one of the methods of recursion.
* It is a straight path in a loop. We can find a solution to solve a problem by using the recursion approach.
* If there is a failure we will go back and think of another approach to solve the problems.

**Steps for backtracking:**

1. Break the problem into subproblems.
2. Solve the sub-problem step by step.
3. If the subproblem doesn't work go to a better approach solution for the given problem.
4. Find the solution until satisfied the given the problem.

**TIME COMPLEXITY:**

* It depends on a number of basic operations of an input function.
* T(0) is 1 because it is a base case.
* T(n)=T(n-1)+3, and this is continuous until T(0) is reached.
* T(n) represents calculating a time complexity.

**Conditions for N-QUEEN problem.**

1. Check rows
2. Check diagonals
3. Check columns.

**BUBBLE SORT:**

**Code:**

* Instead of swapping we use counter to sort the elements.

**CODE:**

def bubble\_sort(arr):

n = len(arr)

for i in range(n):

# Flag to optimize the algorithm

swapped = False

# Last 'i' elements are already in place

for j in range(0, n-i-1):

# Swap if the element found is greater than the next element

if arr[j] > arr[j+1]:

arr[j], arr[j+1] = arr[j+1], arr[j]

swapped = True

# If no two elements were swapped in the inner loop, the array is already sorted

if not swapped:

break

# Example usage:

arr = [64, 34, 25, 12, 22, 11, 90]

bubble\_sort(arr)

print("Sorted array:", arr)