

# Python for Computer Science and Data Science 2 (CSE 3652)

## MINOR ASSIGNMENT-2: COMPUTER SCIENCE THINKING: RECURSION, SEARCHING, SORTING AND BIG O

---

1. Write a recursive function `power(base, exponent)` that, when called, returns

$$base^{exponent}$$

Example: `power(3, 4) = 3*3*3*3`. Both base and exponent are user-defined and assume that the exponent is an integer greater than or equal to 1.

2. The greatest common divisor of integers  $x$  and  $y$  is the largest integer that evenly divides into both  $x$  and  $y$ . Write and test a recursive function `gcd` that returns the greatest common divisor of  $x$  and  $y$ . The gcd of  $x$  and  $y$  is defined recursively as follows: If  $y$  is equal to 0, then  $gcd(x, y)$  is  $x$ ; otherwise,  $gcd(x, y)$  is  $gcd(y, x \% y)$ .
3. Write a recursive function that takes a number  $n$  as an input parameter and prints  $n$ -digit strictly increasing numbers.
4. Implement a recursive solution for computing the  $n$ th Fibonacci number. Then, analyze its time complexity. Propose a more efficient solution and compare the two approaches.
5. Given an array of  $N$  elements, not necessarily in ascending order, devise an algorithm to find the  $k$ th largest one. It should run in  $O(N)$  time on random inputs.
6. For each of the following code snippets, determine the time complexity in terms of Big O. Explain your answer.

(a)

```

1 def example1(n):
2     for i in range(n):
3         for j in range(n):
4             print(i, j)

```

(b)

```

1 for i in range(n):
2     print(i)

```

(c)

```

1 def recursive_function(n):
2     if n <= 1:
3         return 1
4     return recursive_function(n - 1) + recursive_function(n - 1)

```

7. Given  $N$  points on a circle, centered at the origin, design an algorithm that determines whether there are two points that are antipodal, i.e., the line connecting the two points goes through the origin. Your algorithm should run in time proportional to  $N \log N$ .
8. The **quicksort algorithm** is a recursive sorting technique that follows these steps:

**1. Partition Step:** Choose the first element of the array as the pivot and determine its final position in the sorted array by ensuring all elements to its left are smaller and all elements to its right are larger.

**2. Recursive Step:** Recursively repeat the partitioning process on the subarrays created on either side of the pivot.

As an example, consider the array [37, 2, 6, 4, 89, 8, 10, 12, 68, 45] with 37 as the pivot. Using the partitioning logic, the pivot eventually moves to its correct position, resulting in two subarrays: [12, 2, 6, 4, 10, 8] and [89, 68, 45]. The algorithm continues recursively until the entire array is sorted.

Write a Python function `quick_sort` that implements the quicksort algorithm. The function should include a helper function `quick_sort_helper` to handle recursion. The helper function must take a starting and ending index as arguments and sort the array in-place. Demonstrate the function by sorting the given array and printing the sorted output.

9. You are given the following list of famous personalities with their net worth:

- Elon Musk: \$433.9 Billion
- Jeff Bezos: \$239.4 Billion
- Mark Zuckerberg: \$211.8 Billion
- Larry Ellison: \$204.6 Billion
- Bernard Arnault & Family: \$181.3 Billion
- Larry Page: \$161.4 Billion

Develop a program to sort the aforementioned details on the basis of net worth using

- a. Selection sort
- b. Bubble sort
- c. Insertion sort.

The final sorted data should be the same for all cases. After you obtain the sorted data, present the result in the form of the following dictionary:

`{'name1':networth1, 'name2':networth2,...}`

10. Use Merge Sort to sort a list of strings alphabetically. Example:

Input: ['apple', 'orange', 'banana', 'grape']

Output: ['apple', 'banana', 'grape', 'orange']

11. Without using the built-in `sorted()` function, write a Python program to merge two pre-sorted lists into a single sorted list using the logic of Merge Sort. Example:

Input: [1, 3, 5, 7] and [2, 4, 6, 8]

Output: [1, 2, 3, 4, 5, 6, 7, 8]