# Python\_Activity47

"Race of Algorithms – Who's the Fastest, Smartest, and Most Efficient?"

## **6** Goal:

Each group will **explore**, **demonstrate**, and **explain** how a specific algorithm performs in terms of **time and space complexity**, then compare their findings with others.

# Instructions for All Groups:

Each group will:

- 1. Implement a common algorithm in Python.
- 2. **Explain** its time and space complexity.
- 3. **Test** the algorithm on small and large input sizes.
- 4. **Present** their findings to the class (briefly).

#### Group Assignments:

Group	Algorithm	Focus
1	Bubble Sort	Compare time complexity O(n²)
2	Merge Sort	O(n log n) and space trade-off
3	Linear Search	O(n) search vs constant space
4	Binary Search	O(log n) vs sorted list need
5	Fibonacci (Recursive vs DP)	Compare O(2 <sup>n</sup> ) vs O(n)

#### Sample Group Task Breakdown:

Each group should complete:

- 1. Code the algorithm in Python
- 2. Run it with:

- Small input (e.g., 5–10 elements)
- Large input (e.g., 1000+ elements)
- 3. Measure runtime using time module
- 4. **Observe memory** (optional: use sys.getsizeof() or memory\_profiler)
- 5. Answer the following questions in writing or in presentation slides:
  - What is the algorithm's time complexity?
  - What is the space complexity?
  - What did you observe with small vs. large inputs?
  - When is this algorithm useful or not useful?

## **Suggested Output for Presentation:**

Each group can prepare a short (3–5 slide) presentation OR use a poster with:

- Code sample
- Graph/table of runtimes
- Complexity explanation