What is efficiency in programming?

Its basically comparison between Code which is efficient based on TIME and SPACE.

TIME: Which code solve the problem rapidly.

SPACE: Which code contain less memory to solve the same problem.

TIME: Three techniques to measure the time for the code.

1. Measuring Time to execute:

2. Counting Operation involved.

3. Abstract notion of order of growth.

1. MEASURING TIME: using time () function

- ✓ Different time for different algo
  - ☒ Time varies if implementation changes (for into while etc.)
  - ☒ Different machine different time
  - ☒ Does not work for small input
  - ☒ Time varies if input changes, but not establish any relationship

2. Counting Operation: How much operation contain in a code.

- ✓ Different time for different algo.
  - ☒ Time varies if implementation changes (for into while etc.)
- ✓ Different machine different time.
  - ☒ No Clear definition which operation to count.
- ✓ Time varies if input changes, but not establish any relationship.

3. Order of Growth: Relationship between Time(y) and Input(x)

Types of Order or growth:

1. **Constant O (1):**
   Particular item fetching in an Array.
2. **Linear O (n):**
   Linear Search {(Small Array – Less Time) & (Large Array – More Time)}.
3. **Quadratic O (n^2):**
   Nested Loops
4. **Logarithmic O (log n):**
   Binary Search
   Log10 -> if Input (10, 100, 1000) then Time (1,2,3) ➔Input (Multiply) and Time (Add)
   Log2 -> if Input (2,4,8,16) then Time (1,2,3,4)
5. **nlog n O (n log n):**
   Sorting Algo (Merge Sort, Quick Sort etc)
6. **Exponential O(2^n):**
   Reverse of Logarithmic
   If input (1,2,3) then Time (10,100,100) ➔ Input (Add) and Time (Multiply)

Constant > Logarithmic > Linear > nlog n > Quadratic > Exponential