Introduction

Sometimes, a certain problem can be solved in different methods, some of them are efficient than the others. Thus, the is basic subject in computer sciences, where any problem will be solved by designing an algorithm which is converted to a program that aims to achieve the purpose of such problem.

The Difference between the algorithm and the program

An algorithm is a sequence of finite and unambiguous statements for solving a problem, it is a description of how to transform a value, called input, to a value, called output. The description can be expressed in English, pseudo code, or flowcharts.

While the program is an algorithm which is expressed in any programming language.
Characteristics of an algorithm

In any algorithm the following requirements must be established:

1. **Input:** valid inputs are clearly specified or without inputs.
2. **Output:** must be returns at least one output value.
3. **Definiteness:** every statement is clear and unambiguously.
4. **Finiteness:** terminates after a finite number of steps
5. **Effectiveness:** steps are sufficiently simple, basic and can be solved.

Ex: The following algorithm finds the summation of two numbers, it represented in simple nature language.

Step 1: Read the first number in variable x.
Step 2: Read the second number in variable y.
Step 3: Perform the addition of both numbers and store the result in variable z.
Step 4: Return the addition result.

Steps of building good program

1- Requirements Specification: Determine inputs and outputs of the problem.
2- Algorithm Design: Determine basic operations for solving the problem.
3- Algorithm Analysis: A problem can be solved in many methods, these algorithms are analyzed depending on space and time complexities to choose efficient algorithm.
4- Coding: Convert an algorithm to a program after determine suitable data structure for each object.
5- Verification: This step includes three sides:
   - Proving: Prove correctness of the program.
- Testing: Test a program to determine whether faulty results occur (because of errors existence) or not.
- Debugging: Determine positions of program's errors in order to correct them.

**Performance Analysis**

There are two measures that are used in algorithms performance analysis to evaluate them and compare with other algorithms. These measures are:

1) **Space Complexities:**
It is amount of memory that program \( p \) required it during operating until finished.

\[
S(P) = \text{Const.} + S_p \text{ (problem characteristics)}
\]

where (\text{const.} is constant space which is independent on instance characteristics), (\( S_p \) is inconstant space which is dependent on instance characteristics).

2) **Time Complexities:**
Required time for operating the program until finished.

\[
T(P) = \text{Const.} + T_p \text{ (problem characteristics)}
\]

Where:
- \( \text{Const} \) : Compiler time
- \( T_p \) (problem characteristics) : running time
Importantly of Complexities Computing

Space complexities are computed for the following reasons:
1. If the computer is used by multi user.
2. To know if there is enough memory for executing the program.
3. Select the best solution among many solutions that it requires few space.

Time complexities are computed for the following reasons:
1. If the program is executed in real time systems.
2. If there are many solutions of the problem, the selection is done dependently on the an expected performance of them.
3. Some systems make the user to execute the program in limited time.

Computing the program's complexities

Space complexity:

It is computed by counting number of cells which are required to achieve problem steps, without computing storage of algorithm's inputs.

Time complexity:

There are two methods are used to compute and estimate the time complexities.

1- Operation counts:
   By selection one or more operation like addition, subtraction, assignment, …

2- Step counts:
   By counting the total number of running steps in algorithm.
**Note:** The time of a certain step in a program is independent on instance characteristics, like 10 operations of summation are counted as one step, and 100 operations of summation are counted as one step, but $n$ operations of summation (where $n$ is the instance characteristics) are not counted as one step.