4) Scalability ⇒

Algorithms can handle large amounts of data, and can scale up to handle increasingly complex problems.

5) Language independence ⇒

Algorithms can be expressed in multiple programming languages, making them language independent. This means that programmers can use the same algorithm in different programming languages.

6) optimization ⇒

Algorithms can be optimized to run faster and use fewer resources. By analyzing the algorithm, programmers can improve the efficiency of the program.

overall, algorithms offer several merits or benefits to programmers, such as accuracy, efficiency, reusability, scalability, language independence, and optimization.

4) Unambiguous →

An algorithm should be unambiguous, meaning that there should be no room for interpretation or ambiguity in the steps.

5) Deterministic →

An algorithm should be deterministic, meaning that it should produce the same output for a given input every time it is executed.

6) Feasibility →

An algorithm should be feasible, meaning that it should be possible to execute the algorithm in a reasonable amount of time and with a reasonable amount of resources.

7) Language independent →

An algorithm should be language independent, meaning that it can be expressed in multiple programming languages.