Algorithms

What is an Algorithm?

Computer scientist Niklaus Wirth stated that

Program = Algorithms + Data

An algorithm is a part of the plan for the computer program.

An algorithm is ‘an effective procedure for solving a problem in a finite number of steps’.

It is effective, which means that an answer is found and it has a finite number of steps.

A well-designed algorithm will always provide an answer; it may not bethe desired answer but there will be an answer.

It may be that the answer is that there is no answer. A well- designed algorithm is also guaranteed to terminate.

Algorithms and Humans

Algorithms are not a natural way of stating a problem’s solution, because we do not normally state our plan of action.

- We tend to *execute as we think* about the problem. Hence, there are inherent difficulties when writing an algorithm.
- We normally *tailor our plans of action to the particular problem at hand* and not to a general problem (i.e. a nearsighted approach to problem solving)

Algorithms and Humans

We *usually do not write out our plan*, because we are usually unaware of the basic ideas we use to formulate the plan. We hardly think about it – we just do it.

Computer programmers need to adopt a *scientific approach* to problem solving, i.e. writing algorithms that are comprehensive and precise.

We need to be aware of the *assumptions* we make and of the initial conditions.

Algorithms and Humans

Be careful not to *overlook a step* in the procedure just because it seems obvious.

Remember, *machines* do not have judgment, intuition or common sense!

Algorithms

- In mathematics, computer science, and related subjects,
- An *algorithm* is a finite sequence of steps expressed for solving a problem.
- An *algorithm* can be defined as “a process that performs some sequence of operations in order to solve a given problem”.
- Algorithms are used for calculation, data processing, and many other fields.

Algorithms

- Three reasons for using algorithms are *efficiency*, *abstraction* and *reusability*.

Efficiency:

- Certain types of problems, like sorting, occur often in computing.
- Efficient algorithms must be used to solve such problems considering the time and cost factor involved in each algorithm.

Algorithms

- Three reasons for using algorithms are *efficiency*, *abstraction* and *reusability*.

Abstraction:

- Algorithms provide a level of abstraction in solving problems because many seemingly complicated problems can be distilled into simpler ones for which well known algorithms exist.
- Once we see a more complicated problem in a simpler light, we can think of the simpler problem as just an abstraction of the more complicated one.
- For example, imagine trying to find the shortest way to route a packet between two gateways in an internet.
- Once we realize that this problem is just a variation of the more general shortest path problem, we can solve it using the generalised approach.
-

Algorithms

- Three reasons for using algorithms are *efficiency*, *abstraction* and *reusability*.

Reusability:

- Algorithms are often reusable in many different situations.
- Since many well-known algorithms are the generalizations of more complicated ones,
- Many complicated problems can be distilled into simpler ones,
- An efficient means of solving certain simpler problems potentially lets us solve many complicated problems.

Different Ways of Stating Algorithms

- Algorithms may be represented in various ways.
  - Step-form
    - In the step form representation, the procedure of solving a problem is stated with written statements. Each statement solves a part of the problem and these together complete the solution.
  - Pseudo-code
    - is a written form representation of the algorithm. the pseudo-code, which is in human language, tends toward more precision by using a limited vocabulary.

Different Ways of Stating Algorithms

- Algorithms may be represented in various ways.
  - Flowchart
    - are graphically oriented representation forms.
    - They use symbols and language to represent sequence, decision, and repetition actions.

Benefits of using Algorithms

- The use of algorithms provides a number of benefits.
- One of these benefits is in the *development of the procedure* itself, which involves identification of the processes, major decision points, and variables necessary to solve the problem.
- Developing an algorithm allows and even forces examination of the solution process in a rational manner.
- Identification of the processes and decision points reduces the task into a series of smaller steps of more manageable size.
- Problems that would be difficult or impossible to solve in entirety can be approached as a series of small, solvable sub-problems.

FLOWCHARTS

- A *Flowchart* is a type of diagram (graphical or symbolic) that represents an algorithm or process.
- Each step in the process is represented by a different symbol and contains a short description of the process step.
- The flow chart symbols are linked together with arrows showing the process flow direction. A flowchart typically shows the flow of data in a process, detailing the operations/steps in a pictorial format which is easier to understand than reading it in a textual format.
- A *flowchart* describes what operations (and in what sequence) are required to solve a given problem.
- Flowcharts are a pictorial or graphical representation of a process.
- The purpose of all flow charts is to communicate how a process works

Advantages of Using Flowcharts

- *Communication*: Flowcharts are better way of communicating the logic of a system to all concerned.
- *Effective analysis*: With the help of flowchart, problem can beanalysed in more effective way.
- *Proper documentation*: Program flowcharts serve as a goodprogram documentation, which is needed for various purposes.
- *Efficient Coding*: The flowcharts act as a guide or blueprint during the systems analysis and program development phase.
- *Proper Debugging*: The flowchart helps in debugging process.
- *Efficient Program Maintenance*: The maintenance of operating program becomes easy with the help of flowchart. It helps the programmer to put efforts more efficiently on that part.

Limitations of Using Flowcharts

- *Complex logic*: Sometimes, the program logic is quite complicated. In that case, flowchart becomes complex and clumsy.
- *Alterations and Modifications*: If alterations are required the flowchart may require re-drawing completely.
- *Reproduction*: As the flowchart symbols cannot be typed, reproduction of flowchart becomes a problem.

Limitations of Using Flowcharts

- *Complex logic*: Sometimes, the program logic is quite complicated. In that case, flowchart becomes complex and clumsy.
- *Alterations and Modifications*: If alterations are required the flowchart may require re-drawing completely.
- *Reproduction*: As the flowchart symbols cannot be typed, reproduction of flowchart becomes a problem.