

BUILDING BLOCKS OF ALGORITHM

• What is Algorithm ?

An algorithm is a sequence of steps that solves a problem. While algorithms are a fundamental concept in computer science, computer scientists have not agreed on a formal definition. There are many competing definitions, but Donald Knuth's is among the best known. He describes an algorithm as a definite, effective, and finite process that receives input and produces output based on this input. A common addition to this list is correctness. For example: Computing approx value of π (Pi).

• Building Blocks of Algorithm.

In Computer programming, a statement is the smallest standalone element of an imperative Programming language that expresses some action to be carried out. It is an instruction written in a high-level language that commands the computer to perform a specified action. A program written in such a language is formed by a sequence of one or more statements. A statement may have internal components (e.g., expressions).

Kinds of statements

i) Simple statements

- * assignment : $A := A + 3$
- * goto : goto next;
- * return : return s ;
- * call : function (\dots)

ii) Compound statements.

- * block : begin ---- end
- * do-loop : do ---- while ($i < 10$);

* for-loop: for (...) _____ * if-statement: if ---
* else --- * switch-statement; * while-loop.

i) State

An algorithm is deterministic automation for accomplishing a goal which, given an initial state, will terminate in a defined end-state.

Typically, when an algorithm is associated with processing information, data is read from an input source or device, written to an output sink or device, and/or stored for further processing. Stored data is regarded as part of the internal state of the entity performing the algorithm. In practice, the state is stored in data structure.

ii) Control flow

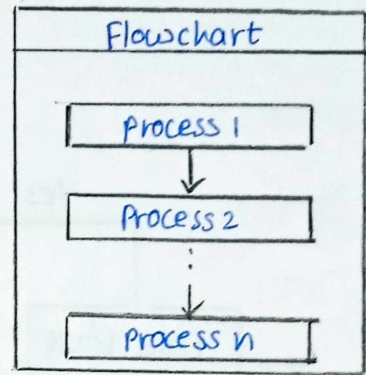
Alternatively referred as flow of control, control flow (when referring to computer programming) is the order of instructions, statements and functions call are executed or evaluated when a program is running. Many programming languages have what are called control flow statements; used to determine what section of code is run in a program at a given time.

a) Sequence Control Structure

As the name suggest the sequential control structure is used to perform the actions one after another. it performs process. A and then perform B process and so on. This structure is represented by writing one process after another. The logic flow is top to bottom approach.

E.g.:

1. Start
2. Get input for A and B
3. Calculate $C = A * B$
4. Print the value of C
5. Stop

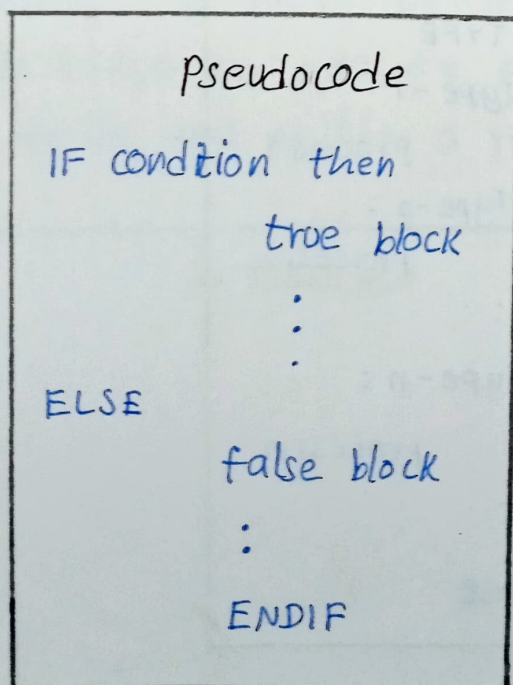


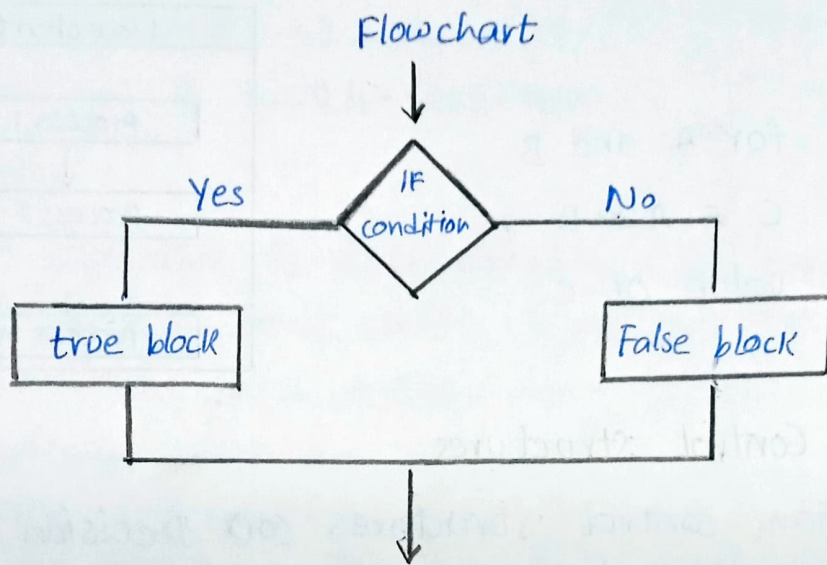
ii) Selection Control structures

Selection control structures (or) Decision structures allows the program to make a "choice between two alternate paths" whether is true or false. IF...THEN or IF...THEN...ELSE or a case structures are the selection structures. This logic is used for making decisions.

a) IF...THEN structures

In this structure if the condition is true it executes true block else if the condition is false it executes false block. In this either the process 1 or process 2 gets executed depending on the condition.





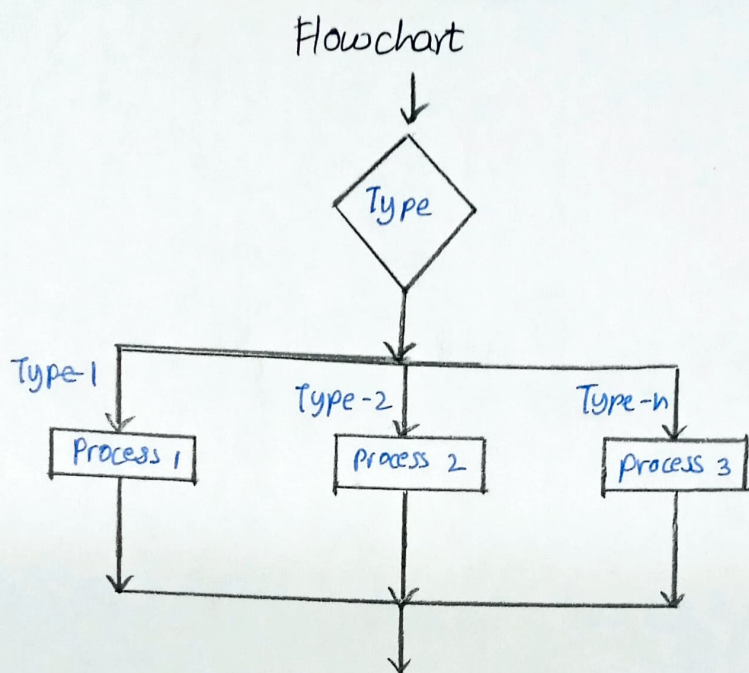
iii) Case structure

This is a "multiway selection structures" that is used to choose one option from many options. if the value of type is equal to Type-1 then it executes process 1, If it is equal to Type-2 it executes process 2 and so on. ENDCASE is used to indicate the end of the CASE structure.

Pseudo code

```

CASE TYPE
case Type-1 :
    process 1
case Type-2 :
    process 2
    ::
case Type-n :
    process n
    ::
END CASE
  
```



iii) statement

statement is a sequence of instruction to accomplish the task or solve a problem. An instruction describes an action. When the instructions are executed a process evolves which accomplishes the intended task (or) solve the problem. An algorithm consists of a finite number of statements. It must be in a ordered form. Time taken to execute all the statement should be finite and within a reasonable limit.

~ Thankyou