CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

DATA STRUCTURE
(CAT-201)

Design By:
Ms. Gurpreet kaur dhiman
Ms.Mandeep kaur
Chandigarh University-Gharuan

Algorithm

Algorithm is a step-by-step procedure, which defines a set of instructions to be executed in a certain order to get the desired output. Algorithms are generally created independent of underlying languages, i.e. an algorithm can be implemented in more than one programming language.

- **From the data structure point of view, following are some important categories of algorithms −**
- **Search** − Algorithm to search an item in a data structure.
- **Sort** − Algorithm to sort items in a certain order.
- **Insert** − Algorithm to insert item in a data structure.
- **Update** − Algorithm to update an existing item in a data structure.
- **Delete** − Algorithm to delete an existing item from a data structure.

Algorithmic Notation

CU
CHANDIGARH UNIVERSITY

The format for the formal presentation of an algorithm consists of two parts. The first part is a paragraph which tells the purpose of the algorithm, identifies the variables which occur in the algorithm and lists the input data. The second part of the algorithm consists of the lists of steps that is to be executed.

Example : A nonempty array DATA with N numerical values is given. Find the location LOC and the value MAX of the largest element of DATA.

Algorithmic Notation

Algorithm 2.3: Given a nonempty array DATA with N numerical values, this algorithm finds the location LOC and the value MAX of the largest element of DATA.

1. Set K : = 1, LOC : =1 and MAX : =DATA[1].
2. Repeat steps 3 and 4 while K<=N:
3. If MAX<DATA[K], then :

   Set LOC : =K and MAX : =DATA[K].

   [End of if structure]
4. Set K : = K+1.
5. Write: LOC, MAX.
6. Exit.

Algorithmic Notation

Steps, Control, Exit :

- The steps of the algorithm are executed one after the other, beginning with step 1.
- Control may be transferred to step n by Go to step n
- If several statements appear in the same step, e. g.

  Set K : = 1, LOC : =1 and MAX : =DATA[1].

- Then they are executed from left to right.
- The algorithm is completed when the statement Exit Is encountered.

Comments :

- Step may contain a comment in brackets which indicates the main purpose of the step.

Algorithmic Notation

Variable names :

- Will use capital letters, as in MAX and DATA
- Counters and subscripts will also be capitalized (K, N)
- Assignment Statement :
- Will use the dots-equal notation : =
- MAX : =DATA[1]
- Assigns the value in DATA[1] in MAX.
- Input and Output :
- Data may be input and assigned to variables by means of a read statement
- Read : Variable names.
- Messages placed in quotation marks and Data in variables may be output by Write or print statement.
- Write : Messages and/ or Variable names.

Complexity of Algorithm

The complexity of an algorithm is a function describing the efficiency of the algorithm in terms of the amount of data the algorithm must process. There are two main complexity measures of the efficiency of an algorithm:

- **Time complexity** is a function describing the amount of time an algorithm takes in terms of the amount of input to the algorithm. "Time" can mean the number of memory accesses performed, the number of comparisons between integers, the number of times some inner loop is executed, or some other natural unit related to the amount of real time the algorithm will take.

- **Space complexity** is a function describing the amount of memory (space) an algorithm takes in terms of the amount of input to the algorithm. We often speak of "extra" memory needed, not counting the memory needed to store the input itself.

Complexity of Algorithm

- Suppose M is an algorithm, n size of the input data. The time and space used by the algorithm M are the two main measures for the efficiency of M.
- The time is measured by counting the number of key operations – in sorting and searching algorithms, for example the number of comparisons.
- The space is measured by counting the maximum of memory needed by the algorithm.
- The complexity of an algorithm M is the function f(n) which gives the running time and or storage space requirement of the algorithm in terms of the size n of the input data. Frequently, the storage space required by an algorithm is simply a multiple of the data size n.

Complexity of Algorithm

The two cases one usually investigates in complexity theory are as follows :

1. Worst case : The maximum value of f(n) for any possible input.
2. Average case : The expected value of f(n).

Sometimes we also consider the minimum possible value of f(n), called the best case.Average case assumes a certain probabilistic distribution for the input data.

- Suppose the numbers n1, n2,……,nk occur with respective probabilities p1, p2, . . . Pk .Then the average value E is given by

  E = n1p1 +n2p2+. . . . +nkpk.

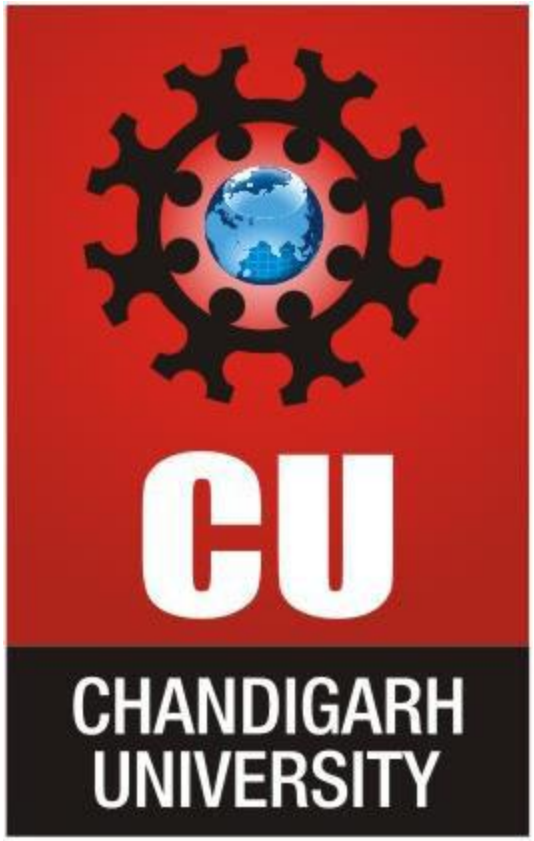
CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

FAQ

- Define algorithm?
- Differentiate three types of complexities.
- Difference b/w space and time complexity.
- Define algorithm notations.

Bibliography

- Seymour Lipschutz, Schaum's Outlines Series Data structures TMH
- Introduction to Data Structures Applications, Trembley&Soreson, Second Edition, Pearson Education
- www.geeksforgeeks.org/analysis-of-algorithms-set-3asymptotic-notations/
- www.tutorialspoint.com/data_structures_algorithms/asymptotic_analysis.htm



Thank You