**Time Complexity Of An Algorithm**

In computer science, the **time complexity** of an **algorithm** quantifies the amount of time taken by an algorithm to run as a function of the size of the input to the problem. The time complexity of an algorithm is commonly expressed using **big O notation**, which suppresses multiplicative constants and lower order terms. When expressed this way, the time complexity is said to be described *asymptotically*, i.e., as the input size goes to infinity. For example, if the time required by an algorithm on all inputs of size *n* is at most 5*n*3 + 3*n*, the asymptotic time complexity is O(*n*3).

Time complexity is commonly estimated by counting the number of elementary operations performed by the algorithm, where an elementary operation takes a fixed amount of time to perform. Thus the amount of time taken and the number of elementary operations performed by the algorithm differ by at most a constant factor.

<http://en.wikipedia.org/wiki/Time_complexity>

Time complexity is the time that an algorithm takes to execute is represented by the number of instructions that are executed in the algorithm .( C++ Classes and Data Structures, Jeffrey S. Childs)

Example:

|  |
| --- |
| namespace TimeComplexity  {  class Program  {  static void Main(string[] args)  {  char[] arr = { 'a', 'b', 'b', 'd', 'e' };  char invalidChar = 'b';  int ptr = 0, N = arr.Length;  for (int i = 0; i < n; i++)  {  if (arr[i] != invalidChar)  {  arr[ptr] = arr[i];  ptr++;  Console.Write(arr[i]);  Console.Write(' ');  }  } |

**Calculate its time complexity**

Let n be the number of elements in the data structure array arr.

We want to express the time complexity in terms of n.

|  |  |
| --- | --- |
| **Statement** | **Time** |
| char[] arr = { 'a', 'b', 'b', 'd', 'e' }; | Will be executed n times |
| char invalidChar = 'b'; | Will be executed 1 time |
| int ptr = 0 | Will be executed 1 time |
| N = arr.Length | Will be executed 1 time |
| int i = 0 | Will be executed 1 time |
| i < n | Will be executed n times |
| i++ | Will be executed n times |
| if (arr[i] != invalidChar) | Will be executed n times |
| arr[ptr] = arr[i]; | Will be executed n times |
| ptr++; | Will be executed n times |
| Console.Write(arr[i]); | Will be executed n times |
| Console.Write(' '); | Will be executed n times |

The overall time = n + 1 + 1 + 1 + 1 + n + n + n + n + n + n + n = 8n + 4

For very large values of n (as n goes to infinity) the value of 4 becomes neglegible, so Time = 8n;