Program Efficiency or Idea of Algorithm Efficiency
Based on CBSE Curriculum
Class -12

By:
Neha Tyagi
PGT CS
KV No-5, 2nd Shift, Jaipur
Algorithm

- An algorithm is a method or procedure for accomplishing a specific task, which is sufficiently precise and that can be performed on computer.
- In Computer science, it is important to measure efficiency of algorithms before applying them in a large scale.
- A good and efficient program starts with an efficient and simple algorithm.
- In different words this is sequence of instructions written in english-like language.
- Performance of an algorithm depends on many internal and external factors-
  - Internal factors:
    (a) Time required to run (b) Space required to run
  - External factors:
    (a) Size of the input to the program (b) Speed of the computer
    (c) Quality of the compiler

इनमे से external factors को कुछ हद तक control किया जा सकता है परन्तु internal factors के लिए algorithm की efficiency को ही measure करना होगा।

Neha Tyagi, KV No-5, Jaipur
What is Computational Complexity?

- It contains two words, one is computation and another one is complexity.
- Computation means to solve problems using an algorithm.
- Complexity involves the study of factors to determine how much resource is sufficiently necessary for this algorithm to run efficiently (performance).
- The Resources generally includes time and space -
  - Time to run the algorithm (temporal complexity)
  - The space (Memory) needed to run the algorithm (Space Complexity)
Algorithm Efficiency in terms of Time

- To determine the efficiency and effectiveness of the algorithm is known as complexity. This is also observed that an algorithm is giving the maximum output in minimum time.

- The Resources generally includes time and space-
  - Time to run the algorithm (temporal complexity)
  - The space (Memory) needed to run the algorithm)(Space Complexity)

- Effectiveness means that the algorithm carries out its Intended Function correctly.

- Efficiency means that the algorithm should be correct with the best possible performance.

- To measure Efficiency we determine complexity.

- Complexity of an algorithm quantifies the resources needed as
Algorithm Efficiency in terms of Time

• Practical Implementation:

```python
#sum of 'n' natural numbers
import time
def sum_n(n):
    start = time.time()  # starting Time
    sum=0
    for i in range(1,n+1):
        sum=sum+1
    end=time.time()
    t=end-start
    print(sum," Time Taken is : ",t,"Seconds")

sum_n(100000)
```

This is clearly visible the difference in time taken in the completion of task. Although the output is same but there is difference in efficiency of both algorithms.

```python
#sum of 'n' natural numbers
import time

start = time.time()  # starting Time
def sum_n2(n):
    return(n*(n+1))/2

print(sum_n2(100000))
end=time.time()     
tag=end-start
print(" Time Taken is : ",t,"Seconds")
```
Algorithm Efficiency on the basis of Number of Operations

• The efficiency of some algorithms depends upon the number of operations which are written to perform a specific task.

• Best examples for this - linear search and binary search

• If we see both algorithms then we come to know that binary search is better because the number of operations in this algorithm are very less. It searches item very fast and in very less operations.

• While in linear search if the item to be searched is the last one in the given list then it will be the worst case. And the item will be found after the number of operations equal to the length of the list.
This is clearly visible the difference in time taken in the completion of task. Although the output is same but there is difference in efficiency of both algorithms. Binary Search is faster than Linear Search because it contains less number of operations.
To increase Efficiency of Python Program

• Keep the Code as compact and simple as possible.
• Looping constructs should be kept to the bare minimum.
• Recursive operations are more efficient than sequential programs with a large number of instructions to be executed.
• Correctness and robustness of the program should be considered first.
• Proper identifier names should be used to increase the readability of a program.
• Best-case, worst-case and average-case should be taken into consideration before writing program/algorithm for a task.
Thank you

Click on the following link for more chapters and material -

www.pythontrends.wordpress.com