ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 9| Issue: 12| December 2023|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2023: 8.224 || ISI Value: 1.188

# ENUMERATE A LIST OF INTEGER ELEMENTS OF A VECTOR USING *enumerate_elements_of_vector_for()* PROCEDURE AND ENUMERATION OF LIST OF INTEGER ELEMENTS USING *enumerate_elements_of_vector_while()* PROCEDURE AND EXAMINING THE TIME COMPLEXITY AND CALCULATING THE SPACE COMPLEXITY OF THE FUNCTIONS OR ALGORITHMS. - A CASE STUDY

**6389 – Cadet P Dheva Dharshan[1]**
*Class- XII 2023-24, Sainik School Amaravathinagar*
*Post: Amaravathinagar, Udumalpet Taluka, Tirupur Dt, Tamilnadu State*

**ABSTRACT**

***In computer science effectiveness of algorithm is exclusively depend on time factor for the execution of included statements within the block of code. Further the amount of memory it is being used for storing data also matters in calculating the space complexity of the program.***

***The 'enumerate_elements_of_vector_for()' procedure employs a 'for' loop to iteratively access and count elements within the vector, while the 'enumerate_elements_of_vector_while()' procedure employs a 'while' loop for the same purpose. Our case study investigates their respective efficiency, taking into account time complexity and space complexity as key metrics.***

***This manuscript specifically examines the time complexity and calculating the space complexity of the for and while functions. By shedding light on the nuances of time and space complexity in the context of enumeration, this case study strives to elucidate best practices for optimizing code and enhancing the overall efficiency of data processing algorithms..***

**KEYWORDS:** ***enumerate_elements_of_vector_for (EVF), Runtime Complexity (rc), Big O𝑂(n), enumerate_elements_of_vector_while() (EVW), Big Theta𝛩(n), Big Omega𝛺(n)***

## 1. INTRODUCTION

A ‘for’ loop is used for iterating over a sequence, that is a list, a tuple, a dictionary, a set or a string. This is less like the ‘for’ keyword in other languages, and works more like an iterator method as found in object-oriented programming languages. With the ‘for’ loop we can execute a set of statements, once for each item in a list, tuple, set etc.[1]

With the ‘while’ loop we can execute a set of statements as long as a condition is true. And when the condition becomes false, the line immediately after the loop in the program is executed. It requires relevant variables to be ready.[4]

## 2. RELATED WORK

All computer languages in the world support sequencing, selection and iteration methods. The syntax and semantics of a language differs because of construction of a compiler or an interpreter is of varied structure.

The construction of a compiler includes the control structures which makes a structured programming language. The ‘for’ and ‘while’ are alternative methodologies for any iteration.
It is experienced by the programmers that the ‘for’ loop is far easier than ‘while’ loop

## 3. METHODOLOGY

To conduct the efficiency comparison, the code generates a dataset in the form of a list named L1.EVF employs a 'for' loop to iteratively go through each element in the list.

A counter variable named v_count is incremented for each element in the list, effectively enumerating the elements. The time taken for enumeration using the 'for' loop is measured by recording the start and end times.

Similar to the 'for' loop method, a 'while' loop is used for enumeration in EVW.

A counter variable, v_count, is incremented during each iteration of the 'while' loop until the end of the list is reached. The time taken for enumeration using the 'while' loop is measured.

The code provides a comparison of the two enumeration methods by evaluating the time taken for each method and the resulting length of the list after enumeration. Comparing these metrics allows you to make an informed decision about the efficiency and performance of 'for' and 'while' loops in this specific context

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 9| Issue: 12| December 2023|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2023: 8.224 || ISI Value: 1.188

enumerate_elements_of_vector_for()and enumerate_elements_of_vector_while().

After running the driver code, the code outputs the length of the list and the time taken for enumeration for both 'for' and 'while' loop methods.

The results allow you to draw conclusions regarding the relative efficiency of these two enumeration methods.

**ALGORITHM ENUMERATE A LIST OF INTEGER ELEMENTS OF A VECTOR USING FOR LOOP (EVF)**
**STEP 01: START**
**STEP 02: INITIALIZE AN EMPTY LIST L1 TO STORE ELEMENTS**
**STEP 03: USING A 'FOR' LOOP, POPULATE THE LIST L1 WITH NUMBERS**
**STEP 04: INITIALIZE A COUNTER VARIABLE V_COUNT TO KEEP TRACK OF THE NUMBER OF ELEMENTS.**
**STEP 05: RECORD THE START TIME OF THE ENUMERATION PROCESS.**
**STEP 06: IN FOR LOOP ADD 1 TO V_COUNT EACH ITERATION**
**STEP 07: RECORD THE END TIME OF THE ENUMERATION PROCESS.**
**STEP 08: DISPLAY THE LENGTH OF L1**
**STEP 09: DISPLAY THE TOTAL TIME TAKEN FOR ENUMERATION**
**STEP 10: STOP**
**PYTHON PROGRAM TO ENUMERATE A LIST OF INTEGER ELEMENTS OF A VECTOR USING FOR LOOP (EVF)**

```
L1=[]
for i in range(1,100):
L1.append(i)
v_count=0
start = time.time()
for i in L1:
 v_count=v_count+1
end = time.time()
print("Length of L1 using For Loop=",v_count)
print("Total Time For Enumeration Using 'For' Loop: ",end - start)
```

**ALGORITHM TO ENUMERATE A LIST OF INTEGER ELEMENTS OF A VECTOR USING WHILE LOOP (EVW)**
**STEP 01: START**
**STEP 02: INITIALIZE AN EMPTY LIST L1 TO STORE ELEMENTS**
**STEP 03: USING A LOOP, POPULATE THE LIST L1 WITH NUMBERS**
**STEP 04: INITIALIZE A COUNTER VARIABLE V_COUNT TO KEEP TRACK OF THE NUMBER OF ELEMENTS.**
**STEP 05: INITIALIZE A VARIABLE ELE WITH THE NUMBER OF ELEMENTS USING IN BUILT FUNCTION**
**STEP 06: RECORD THE START TIME OF THE ENUMERATION PROCESS.**
**STEP 07: IN WHILE LOOP ADD 1 TO V_COUNT EACH ITERATION WHILE i IS LESS THAN ELE**
**STEP 08: RECORD THE END TIME OF THE ENUMERATION PROCESS.**
**STEP 09: DISPLAY THE LENGTH OF L1**
**STEP 10: DISPLAY THE TOTAL TIME TAKEN FOR ENUMERATION**
**STEP 11: STOP**

**PYTHON PROGRAM TO ENUMERATE A LIST OF INTEGER ELEMENTS OF A VECTOR USING WHILE LOOP (EVW)**

```
L1=[]
for i in range(1,100):
  L1.append(i)
ele=len(L1)
i=0
v_count=0
start = time.time()
while(i<ele):
 v_count=v_count+1
  i=i+1
end = time.time()
print("Length of L1 using 'While'      Loop=",v_count)
  print("Total Time For Enumeration Using 'While' Loop: ",end - start)
```

## 4. COMPLEXITY OF ALGORITHM

In computer science, analysis of algorithms is a very crucial part. It is important to find the most efficient algorithm for solving a problem. It is possible to have many algorithms to solve a problem, but the challenge here is to choose the most efficient one. [2]

There are multiple ways to design an algorithm, or considering which one to implement in an application. When thinking through this, it's crucial to consider the algorithm's **time complexity** and **space complexity**. [3]

## 5. SPACE COMPLEXITY

The space complexity of an algorithm is the amount of space (or memory) taken by the algorithm to run as a function of its input length, n. Space complexity includes both auxiliary space and space used by the input.[3]

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 9| Issue: 12| December 2023|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2023: 8.224 || ISI Value: 1.188

Auxiliary space is the temporary or extra space used by the algorithm while it is being executed. Space complexity of an algorithm is commonly expressed using **Big (O(n))** notation.[3]

The Space complexity is ignored in this research paper, since the space complexity of particular problem is not considered so important.

## 6. TIME COMPLEXITY

The time complexity of an algorithm is the amount of time taken by the algorithm to complete its process as a function of its input length, n. The time complexity of an algorithm is commonly expressed using asymptotic notations:[3]

**Big O - O(n)**
**Big Theta - Θ(n)**
**Big Omega - Ω(n)**

It's valuable for a programmer to learn how to compare performances of different algorithms and choose the best time-space complexity to solve a particular problem in the most efficient way possible. [3]

**Big O** notation is used in Computer Science to portrait the performance or complexity of an algorithm.

**Big O** specifically defines the worst-case scenario of an algorithm, and can be used to describe the execution time required or the space used (e.g. in memory or on disk) by an algorithm. here O stands for order of growth.

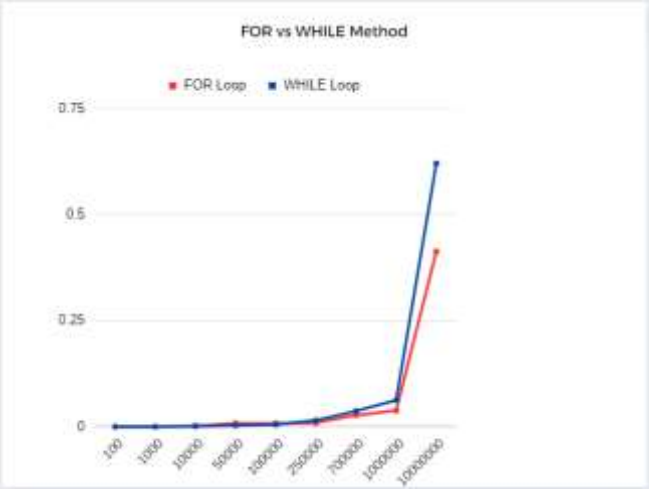
**Big Theta(Θ)** is used to represent the average case scenario of an algorithm and can be used to describe the execution time required or the space used (e.g. in memory or on disk) by an algorithm.

**Big Omega (Ω)**is used to represent the best case scenario of an algorithm and can be used to describe the execution time required or the space used (e.g. in memory or on disk) by an algorithm.

These three methods are the most common and very popular methods of design and analysis of an algorithm which are used for finding the efficiency of the program.

## 7. RUNTIME COMPLEXITY OF ENUMERATION OF A VECTOR

| No. of Elements | Using EVF | Using EVW |
|---|---|---|
| **100** | 0 | 0 |
| **1000** | 0 | 0 |
| **10000** | 0.00199461 | 0.000994205 |
| **50000** | 0.00794816 | 0.002991438 |
| **100000** | 0.007977724 | 0.004986525 |
| **250000** | 0.008976698 | 0.014961004 |
| **700000** | 0.026955366 | 0.0369828892 |
| **1000000** | 0.03824217 | 0.062833071 |
| **10000000** | 0.412894726 | 0.620584965 |




***Graphical Representation of Runtime complexity of both the methods***

## 8.'WHILE' LOOP METHOD – *rc*

In the EVW the program enumerate the list of integers, the time complexity of the algorithm for worst case is denoted as:

**Big (O(n))**

## 9.'FOR' LOOP METHOD – *rc*

The time complexity of the EVF is calculated as

**Big (O(n))**

## 10. CONCLUSION

The EVF has the greater efficiency for counting when comparing EVW for large number of elements while EVW is more efficient for less number of elements. Further it is also observed that generating integers and storing in a list is one time process and it is time consuming but once the list is prepared the performance of EVF is much higher than the EVW. In addition to this it is also observed that the execution of expression also depends on the hardware configuration. The space complexity for the EVF and EVW is identical.

## 11.ACKNOWLEDGEMENT

Apart from the efforts of me, the success of any work or project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this research paper.

I express deep sense of gratitude to almighty God for giving me strength for the successful completion of the research paper.

I express my heartfelt gratitude to my parents for constant encouragement while carrying out this research paper
.

I express my deep sense of gratitude to **The Principal Capt. (IN) K Manikamdan, Sainik School Amaravathinagar** who has been continuously motivating and extending their helping hand to us.

I express my sincere thanks to the academician The **Vice Principal Wg Cdr Deepti Upadhyaya, Sainik School Amaravathinagar**, for constant encouragement and the guidance provided during this research.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 9| Issue: 12| December 2023|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2023: 8.224 || ISI Value: 1.188

I express my earnest thanks to the academician The **Administrative Officer Lt Col Deepu, Sainik School Amaravathinagar**, for constant reassurance and the guidance provided during this research.

My sincere thanks to **Mr. Praveen Kumar Murigeppa Jigajinni**, Master In-charge. A guide, Mentor and great motivator, who critically reviewed my paper and helped in solving each and every problem, occurred during implementation of this research paper.

## 12. REFERENCES

1. *https://www.w3schools.com/python/python_for_loops.asp*
2. *https://www.freecodecamp.org/news/time-complexity-of-algorithms/*
3. *https://www.educative.io/edpresso/time-complexity-vs-space-complexity*
4. *https://www.w3schools.com/python/python_while_loops.asp*