

**“AZƏRBAYCAN HAVA YOLLARI” CJSC NATIONAL AVIATION ACADEMY**

**Individual Work № 2:**

**Topic: Space Complexity**

**Subject: Obyektyönümlü proqramlaşdırma**

**Teacher: Mamed Shahmaliyev**

**Group: 1459i Student: Azadaliyev Rustam**

**Date: 11.10.2021 Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_**

**Space Complexity:**The **space complexity** of an algorithm quantifies the amount of space taken by an algorithm to run as a function of the length of the input. Consider an example: Suppose a problem to find the frequency of array elements

The pseudo-code is as follows:

int freq[n];

int a[n];

for(int i = 0; i<n; i++)

{

cin>>a[i];

freq[a[i]]++;

}

Below is the implementation of the above approach:

|  |
| --- |
| def countFreq(arr, n):      freq = dict()      for i in arr:          if i not in freq:              freq[i] = 0          freq[i]+=1      for x in freq:          print(x, freq[x])  arr =  [10, 20, 20, 10, 10, 20, 5, 20 ]  n = len(arr)  countFreq(arr, n) |

**Output**

5 1

10 3

20 4

Here two arrays of length **N**, and variable **i** are used in the algorithm so, the total space used is **N \* c + N \* c + 1 \* c = 2N \* c + c**, where **c** is a unit space taken. For many inputs, constant **c** is insignificant, and it can be said that the space complexity is **O(N)**.

There is also **auxiliary space,**which is different from space complexity. The main difference is where space complexity quantifies the total space used by the algorithm, auxiliary space quantifies the extra space that is used in the algorithm apart from the given input. In the above example, the auxiliary space is the space used by the freq[] array because that is not part of the given input. So total auxiliary space is **N \* c + c** which is **O(N)** only.