

# Select a random number from stream, with $O(1)$ space

Difficulty Level : Hard • Last Updated : 21 Jan, 2022



Given a stream of numbers, generate a random number from the stream. You are allowed to use only  $O(1)$  space and the input is in the form of a stream, so can't store the previously seen numbers.

So how do we generate a random number from the whole stream such that the probability of picking any number is  $1/n$ . with  $O(1)$  extra space? This problem is a variation of [Reservoir Sampling](#). Here the value of  $k$  is 1.

**1)** Initialize 'count' as 0, 'count' is used to store count of numbers seen so far in stream.

**2)** For each number 'x' from stream, do following

.....**a)** Increment 'count' by 1.

.....**b)** If count is 1, set result as x, and return result.

.....**c)** Generate a random number from 0 to 'count-1'. Let the generated random number be i.

.....**d)** If i is equal to 'count - 1', update the result as x.

```

#include <bits/stdc++.h>
#include <time.h>
using namespace std;

// A function to randomly select a item
// from stream[0], stream[1], .. stream[i-1]
int selectRandom(int x)
{
    static int res; // The resultant random number
    static int count = 0; // Count of numbers visited
                        // so far in stream

    count++; // increment count of numbers seen so far

    // If this is the first element from stream,
    // return it
    if (count == 1)
        res = x;
    else
    {
        // Generate a random number from 0 to count - 1
        int i = rand() % count;

        // Replace the prev random number with
        // new number with 1/count probability
        if (i == count - 1)
            res = x;
    }
    return res;
}

// Driver Code
int main()
{
    int stream[] = {1, 2, 3, 4};
    int n = sizeof(stream) / sizeof(stream[0]);

    // Use a different seed value for every run.
    srand(time(NULL));
    for (int i = 0; i < n; ++i)
        cout << "Random number from first " << i + 1

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

// This is code is contributed by rathbhupendra

## C

```
// An efficient C program to randomly select a number from stream of numb
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

// A function to randomly select a item from stream[0], stream[1], .. str
int selectRandom(int x)
{
    static int res;    // The resultant random number
    static int count = 0; //Count of numbers visited so far in stream

    count++; // increment count of numbers seen so far

    // If this is the first element from stream, return it
    if (count == 1)
        res = x;
    else
    {
        // Generate a random number from 0 to count - 1
        int i = rand() % count;

        // Replace the prev random number with new number with 1/count pr
        if (i == count - 1)
            res = x;
    }
    return res;
}

// Driver program to test above function.
int main()
{
    int stream[] = {1, 2, 3, 4};
    int n = sizeof(stream)/sizeof(stream[0]);

    // Use a different seed value for every run.
    srand(time(NULL));
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

```
}
```

## Java

//An efficient Java program to randomly select a number from stream of nu

```
import java.util.Random;
```

```
public class GFG
```

```
{
```

```
    static int res = 0;    // The resultant random number
```

```
    static int count = 0; //Count of numbers visited so far in stream
```

```
    //A method to randomly select a item from stream[0], stream[1], .. st
```

```
    static int selectRandom(int x)
```

```
    {
```

```
        count++; // increment count of numbers seen so far
```

```
        // If this is the first element from stream, return it
```

```
        if (count == 1)
```

```
            res = x;
```

```
        else
```

```
        {
```

```
            // Generate a random number from 0 to count - 1
```

```
            Random r = new Random();
```

```
            int i = r.nextInt(count);
```

```
            // Replace the prev random number with new number with 1/coun
```

```
            if(i == count - 1)
```

```
                res = x;
```

```
        }
```

```
        return res;
```

```
    }
```

```
    // Driver program to test above function.
```

```
    public static void main(String[] args)
```

```
    {
```

```
        int stream[] = {1, 2, 3, 4};
```

```
        int n = stream.length;
```

```
        // Driver code to test above function
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

//This code is contributed by Sumit Ghosh

## Python3

```
# An efficient python3 program
# to randomly select a number
# from stream of numbers.
import random

# A function to randomly select a item
# from stream[0], stream[1], .. stream[i-1]
# The resultant random number
res=0
# Count of numbers visited
# so far in stream
count=0
def selectRandom(x):

    global res
    global count

    # increment count of numbers
    # seen so far
    count += 1;

    # If this is the first element
    # from stream, return it
    if (count == 1):
        res = x;
    else:

        # Generate a random number
        # from 0 to count - 1
        i = random.randrange(count);

        # Replace the prev random number
        # with new number with 1/count
        # probability
        if (i == count - 1):
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

```

stream = [1, 2, 3, 4];
n = len(stream);

# Use a different seed value
# for every run.
for i in range (n):
    print("Random number from first",
          (i + 1), "numbers is",
          selectRandom(stream[i]));

# This code is contributed by mits

```

## C#

```

// An efficient C# program to randomly
// select a number from stream of numbers.
using System;

class GFG
{
    // The resultant random number
    static int res = 0;

    // Count of numbers visited
    // so far in stream
    static int count = 0;

    // A method to randomly select
    // a item from stream[0],
    // stream[1], .. stream[i-1]
    static int selectRandom(int x)
    {
        // increment count of
        // numbers seen so far
        count++;

        // If this is the first
        // element from stream,
        // return it
        if (count == 1)

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

```

        // from 0 to count - 1
        Random r = new Random();
        int i = r.Next(count);

        // Replace the prev random
        // number with new number
        // with 1/count probability
        if(i == count - 1)
            res = x;
    }
    return res;
}

// Driver Code
public static void Main()
{
    int[] stream = {1, 2, 3, 4};
    int n = stream.Length;
    for(int i = 0; i < n; i++)
        Console.WriteLine("Random number from " +
                           "first {0} numbers is {1}" ,
                           i + 1, selectRandom(stream[i]));
}

// This code is contributed by mits

```

## PHP

```

<?php
// An efficient php program to randomly
// select a number from stream of numbers.

// A function to randomly select a item
// from stream[0], stream[1], .. stream[i-1]
function selectRandom($x)
{

    // The resultant random number

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

```

$count = 0;

// increment count of numbers seen
// so far
$count++;

// If this is the first element
// from stream, return it
if ($count == 1)
    $res = $x;
else
{
    // Generate a random number from
    // 0 to count - 1
    $i = rand() % $count;

    // Replace the prev random number
    // with new number with 1/count
    // probability
    if (i == $count - 1)
        $res = $x;
}
return $res;
}

// Driver program to test above function.
$stream = array(1, 2, 3, 4);
$n = sizeof($stream)/sizeof($stream[0]);

// Use a different seed value for
// every run.
srand(time(NULL));

for ($i = 0; $i < $n; ++$i)
    echo "Random number from first ",
        $i+1, "numbers is " ,
        selectRandom($stream[$i]), "\n" ;

// This code is contributed by nitin mittal.
?>

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**



```

<script>
//An efficient Javascript program to randomly select a number from stream

let res = 0;    // The resultant random number
let count = 0; //Count of numbers visited so far in stream

//A method to randomly select a item from stream[0], stream[1], .. stream
function selectRandom(x)
{
    count++; // increment count of numbers seen so far

    // If this is the first element from stream, return it
    if (count == 1)
        res = x;
    else
    {
        // Generate a random number from 0 to count - 1

        let i = Math.floor(Math.random()*(count));

        // Replace the prev random number with new number with 1/count
        if(i == count - 1)
            res = x;
    }
    return res;
}

// Driver program to test above function.
let stream=[1, 2, 3, 4];
let n = stream.length;
for(let i = 0; i < n; i++)
    document.write("Random number from first " + (i+1) + " numbers is " +

// This code is contributed by avanitrachhadiya2155
</script>

```

## Output:

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

Random number from first 3 numbers is 3

Random number from first 4 numbers is 4

**Time Complexity:**  $O(n)$

**Auxiliary Space:**  $O(1)$

### How does this work

We need to prove that every element is picked with  $1/n$  probability where  $n$  is the number of items seen so far. For every new stream item  $x$ , we pick a random number from 0 to 'count - 1', if the picked number is 'count - 1', we replace the previous result with  $x$ .

To simplify proof, let us first consider the last element, the last element replaces the previously-stored result with  $1/n$  probability. So the probability of getting the last element as the result is  $1/n$ .

Let us now talk about the second last element. When the second last element processed the first time, the probability that it replaced the previous result is  $1/(n-1)$ . The probability that the previous result stays when the  $n$ th item is considered is  $(n-1)/n$ . So the probability that the second last element is picked in the last iteration is  $[1/(n-1)] * [(n-1)/n]$  which is  $1/n$ .

Similarly, we can prove for third last element and others.

References:

[Reservoir Sampling](#)

**Method 2:** generate a random number from the stream with numpy `random.choice()` method.

---

## Python3

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

```
# using random.choice() to
# get a random number
random_num = np.random.choice(stream)

# printing random number
print("random number is ",random_num)
```

### Output:

7

**Time Complexity:**  $O(n)$

**Auxiliary Space:**  $O(1)$

**LEVEL UP YOUR COMPETITIVE PROGRAMMING GAME**

Mathematical Concepts | Live Classes | Doubt Support

Enrol Now



GeeksforGeeks

Like 13

Previous

**Reservoir Sampling**

Next

**Find the largest multiple of 2,  
3 and 5**

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

**01** Select a Random Node from a tree with equal probability  
14, Jan 18

**05** Random number generator in arbitrary probability distribution fashion  
06, Jun 13

**02** Select a Random Node from a Singly Linked List  
24, May 15

**06** Pseudo Random Number Generator (PRNG)  
27, Jun 17

**03** Implement random-0-6-Generator using the given random-0-1-Generator  
18, Jan 18

**07** Probability of getting a perfect square when a random number is chosen in a given range  
07, May 20

**04** Number of ways to select exactly K even numbers from given Array  
18, Jul 20

**08** Java Program to Implement Park-Miller Random Number Generation Algorithm  
13, Dec 20

**Article Contributed By :**



**GeeksforGeeks**

**Vote for difficulty**

Current difficulty : [Hard](#)

Easy

Normal

Medium

Hard

Expert

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

**Improved By :**    [nitin mittal](#), [Mithun Kumar](#), [rathbhupendra](#), [nidhi\\_biet](#),  
[avanitrachhadiya2155](#), [pulamolusaimohan](#),  
[subhammahato348](#), [prathamgoyal226](#)

**Article Tags :**    [array-stream](#), [Random Algorithms](#), [Mathematical](#),  
[Randomized](#)

**Practice Tags :**    [Mathematical](#)

Improve Article

Report Issue

Writing code in comment? Please use [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), generate link and share the link here.

Load Comments

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**



A-143, 9th Floor, Sovereign Corporate Tower,  
Sector-136, Noida, Uttar Pradesh - 201305

[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

## Company

[About Us](#)  
[Careers](#)  
[In Media](#)  
[Contact Us](#)  
[Privacy Policy](#)  
[Copyright Policy](#)

## Learn

[Algorithms](#)  
[Data Structures](#)  
[SDE Cheat Sheet](#)  
[Machine learning](#)  
[CS Subjects](#)  
[Video Tutorials](#)  
[Courses](#)

## News

[Top News](#)  
[Technology](#)  
[Work & Career](#)  
[Business](#)  
[Finance](#)  
[Lifestyle](#)  
[Knowledge](#)

## Languages

[Python](#)  
[Java](#)  
[CPP](#)  
[Golang](#)  
[C#](#)  
[SQL](#)  
[Kotlin](#)

## Web Development

## Contribute

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

---

NodeJS

@geeksforgeeks , Some rights reserved

Do Not Sell My Personal Information

---

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**