

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

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 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.

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

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

    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
        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));
    for (int i = 0; i < n; ++i)
        printf("Random number from first %d numbers is %d\n",
               i+1, selectRandom(stream, i));

    return 0;
}
```

Output:

```
Random number from first 1 numbers is 1
Random number from first 2 numbers is 1
Random number from first 3 numbers is 3
Random number from first 4 numbers is 4
```

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 probability of getting last element as result is  $1/n$ .

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