



Kyle Lang &lt;kyle.everett.lang@gmail.com&gt;

---

## How to Pick a Random Element from an Infinite Stream

1 message

---

**Daily Coding Problem** <founders@dailycodingproblem.com>

Sun, Sep 8, 2019 at 5:21 PM

To: kyle.everett.lang@gmail.com

**Daily Coding Problem**

---

Hey there,

Hope your interview practicing is going well! Let's work through the problem of uniformly picking a random element from a gigantic stream. This is a common interview question at companies like Google and Facebook.

Naively, we could process the stream and store all the elements we encounter in a list, find its size, and pick a random element from `[0, size - 1]`. The problem with this approach is that it would take  $O(N)$  space for a large  $N$ .

Instead, let's attempt to solve using loop invariants. On the  $i$ th iteration of our loop to pick a random element, let's assume we already picked an element uniformly from `[0, i - 1]`. In order to maintain the loop invariant, we would need to pick the  $i$ th element as the new random element at  $1 / (i + 1)$  chance. For the base case where `i = 0`, let's say the random element is the first one. Then we know it works because

- For `i = 0`, we would've picked uniformly from `[0, 0]`.
- For `i > 0`, before the loop began, any element `k` in `[0, i - 1]` had  $1 / i$  chance of being chosen as the random element. We want `k` to have  $1 / (i + 1)$  chance of being chosen after the iteration. This is the case since the chance

of having being chosen already but not getting swapped with the  $i$ th element is  $1 / i * (1 - (1 / (i + 1)))$  which is  $1 / i * i / (i + 1)$  or  $1 / (i + 1)$

Let's see how the code would look:

```
import random

def pick(big_stream):
    random_element = None

    for i, e in enumerate(big_stream):
        if i == 0:
            random_element = e
        if random.randint(1, i + 1) == 1:
            random_element = e
    return random_element
```

Since we are only storing a single variable, this only takes up constant space!

Turns out this algorithm is called [Reservoir Sampling](#). If you liked this problem and solution, sign up for our mailing list below for problems like these! You can also read our other post on how to solve tricky interview questions like this one [here](#).

Best of luck!

Marc

If you liked this guide, feel free to forward it along! As always, shoot us an email if there's anything we can help with!

---

[Unsubscribe.](#)