Reservoir Sampling

Sand Ip

LeetCode

382. Linked List Random Node

Medium 427 133 Favorite Share

Given a singly linked list, return a random node's value from the linked list. Each node must have the **same probability** of being chosen.

Follow up:

What if the linked list is extremely large and its **length is unknown** to you? Could you solve this efficiently without using extra space?

Examine Google's stream of search queries and find a sample of what was asked over the last 24 hours

~63,000 queries per second

~5,443,200,000 queries per day

APPROACH

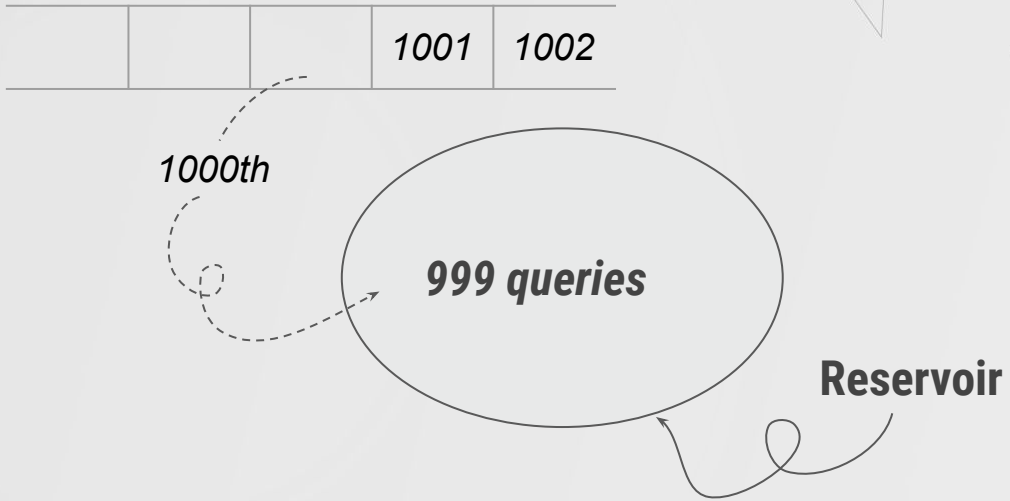
Implement an algorithm that randomly chooses 1000 items from this stream such that each item is equally likely to be selected

ANALYSIS

- Data stream is large and will not fit into main memory
- Can only iterate over data once
- Unbiased selection of an item to be placed in sample

STEP 1

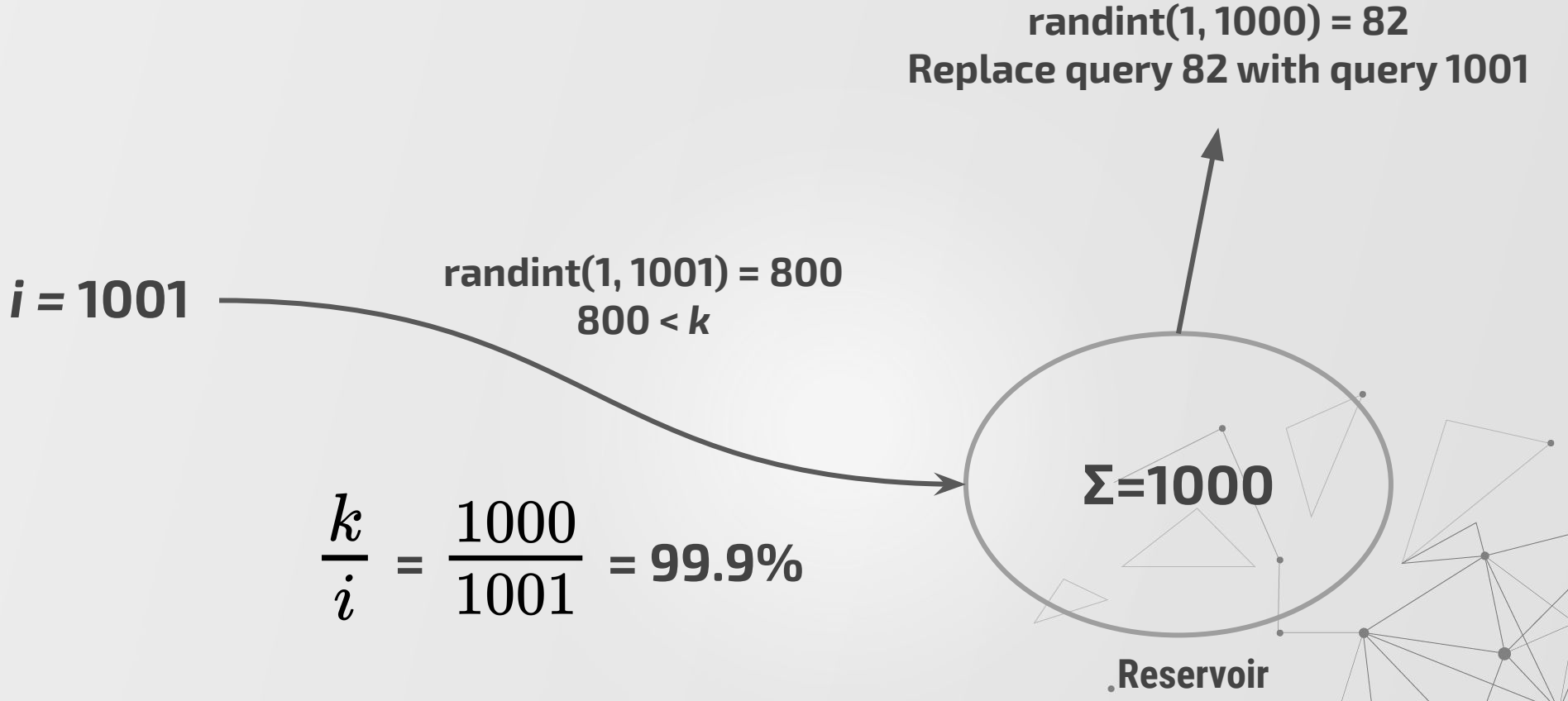
The first *k* events in the stream automatically enters the reservoir



STEP 2

For the ***i***th event if ***i*** > ***k***, draw a random number, ***q***, between 0 and ***i***.

If ***q*** is smaller than ***k*** (sample size=1000), randomly replace an element in the reservoir with the *i*th element


randint(1, 1000) = 82
Replace query 82 with query 1001
i = 1001
randint(1, 1001) = 800
800 < k
Σ=1000
Reservoir
$\frac{k}{i} = \frac{1000}{1001} = 99.9\%$

$i = 10{,}000$

randint(1, 10000) = 3100

3100 < k

Σ=1000

$$\frac{k}{i} = \frac{1000}{10000} = 10\%$$

Reservoir

Over time, as ***i*** increases, the probability of it replacing an existing event in the reservoir decreases


O(n)
O(1) per event

SOURCES

1. https://pycon.org.il/2016/static/sessions/jonathan-arfa.pdf
2. https://www.kdnuggets.com/2019/09/5-sampling-algorithms.html
3. https://gregable.com/2007/10/reservoir-sampling.html
4. https://leetcode.com/problems/linked-list-random-node/


10000
1001