Problem: next-neighbor cannot “jump” to a random potential neighbor of v

Bucketing Divide each row of the adjacency matrix into contiguous buckets

⇒ random neighbor of $v \approx$ random neighbor in a random bucket of v

Problem: Do NOT know $\deg(v)$: Must return each neighbor with prob. $1/\deg(v)$

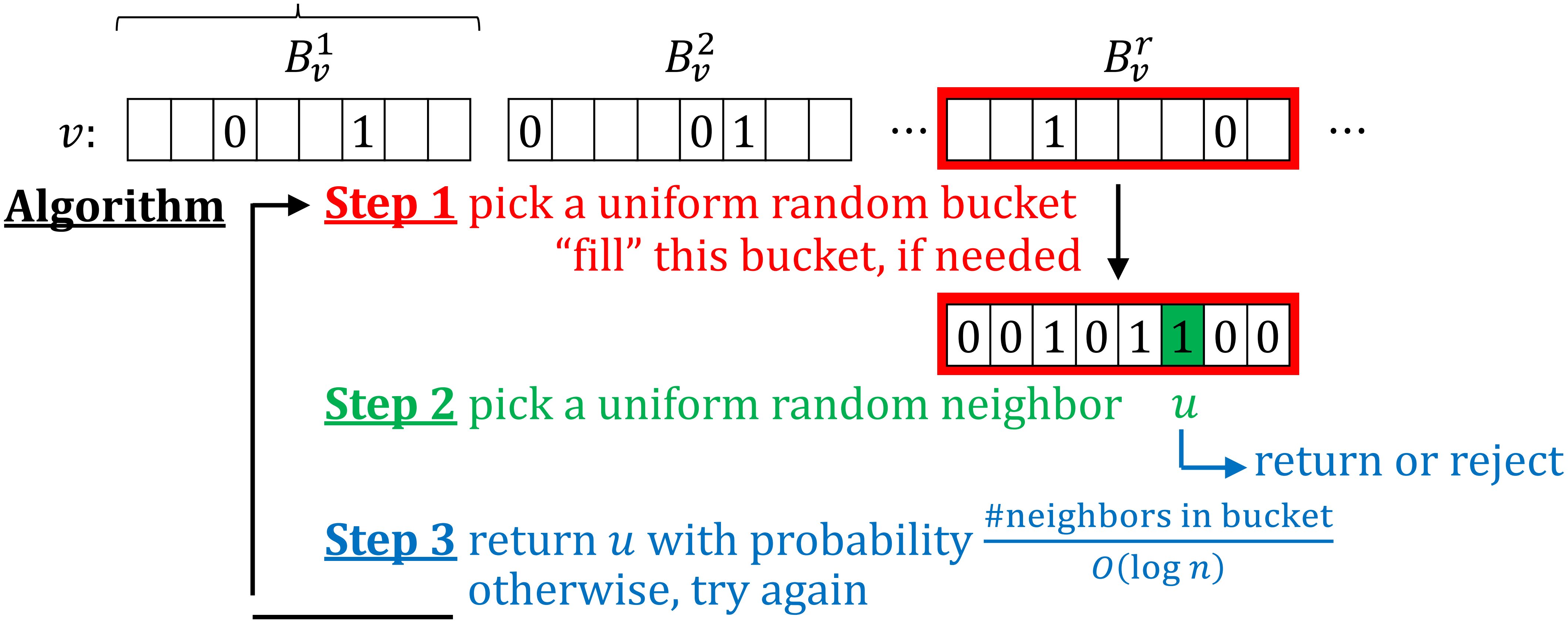
Rejection Sampling Normalize probability of returning any specific neighbor

Problem: next-neighbor cannot “jump” to a random potential neighbor of v

⇒ suffice to show that **any neighbor** is returned with the **equal** probability

#neighbors in each bucket

$\sim \Theta(1)$ in expectation, $O(\log n)$ max w.h.p. ⇒ #buckets ~ #neighbors



Algorithm

Step 1 pick a uniform random bucket

“fill” this bucket, if needed

Step 2 pick a uniform random neighbor u

return or reject

Step 3 return u with probability $\frac{\text{\#neighbors in bucket}}{O(\log n)}$

otherwise, try again

$$\Pr[u \text{ returned}] = \frac{1}{\text{\#buckets}} \times \frac{1}{\text{\#neighbors in bucket}} \times \frac{\text{\#neighbors in bucket}}{O(\log n)} \sim \frac{\Omega(1/\log n)}{\text{\#neighbors}}$$

$\Pr[\text{some neighbor returned}] \sim \Omega(1/\log n) \Rightarrow O(\log n)$ tries suffices

Data Structure Buckets contains set of known neighbors, and **“filled”** marker

⇒ “fill” with expected $\Theta(1)$ next-neighbor queries

⇒ random-neighbor succeeds in $O(\log n)$ tries

$O(\log n)$ time *per query*

$\tilde{O}(m+n)$ space usage