

Exercise walk-through

5.3

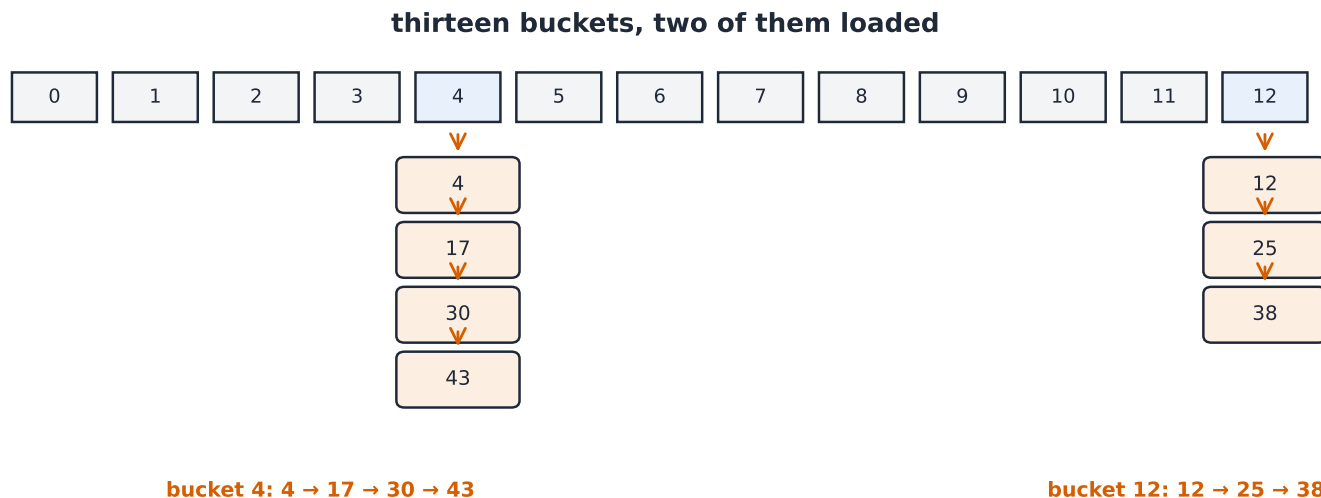
Hashing

University Computer Science

Hashing: spread, collide, resolve

- 1 **Buckets** $h(k) = k \bmod 13$ sends 4, 17, 30, 43 all to bucket 4 - a chain of four; keys 12, 25, 38 fill bucket 12; $\alpha = 7/13$, about 0.54.
- 2 **Probing** Linear probing walks the cluster: $1 + 2 + 3 + 4 = 10$ probes to place those four keys - **clustering** grows the runs.
- 3 **Keep** Real tables hold α under about 0.75 and resize - that is why dictionaries stay fast.

Hash buckets



load factor $\alpha = 7 \text{ keys} / 13 \text{ buckets} \approx 0.54$

Thirteen buckets under $h(k) = k \bmod 13$: keys 4, 17, 30, 43 chain in bucket 4 and 12, 25, 38 chain in bucket 12.

Exercises 2.1-2.3

2.1 Give $h(k)$ for $k = 29, 55, 68$ under mod 13. [3]

2.2 Twelve keys in thirteen buckets: state α and the average chain length. [3]

2.3 State why a prime modulus beats a power of two when keys share low bits. [3]

Exercises 2.1-2.3

2.1 Give $h(k)$ for $k = 29, 55, 68$ under mod 13. [3]

2.1

- $29 \bmod 13 = 3$; $55 \bmod 13 = 3$; $68 \bmod 13 = 3$ ✓.
-

2.2 Twelve keys in thirteen buckets: state α and the average chain length. [3]

2.2

- $\alpha = 12/13 \approx 0.92$ ✓.
 - The average chain length equals $\alpha \approx 0.92$ ✓.
-

2.3 State why a prime modulus beats a power of two when keys share low bits. [3]

2.3

- A power-of-two modulus keeps only the low bits – keys sharing them collide wholesale ✓.
- A prime spreads the residue across all bits of the key ✓.

Exercise 3.1

3.1 A hash table has $m = 11$ buckets with chaining, and the keys 6, 17, 28, 39, 50 arrive in order.

Compute each key's bucket, and give every non-empty chain after all insertions. State α , the expected successful-search cost $1 + \alpha/2$, and the exact worst-case comparison count for searching 50 in your table. [8]

Exercise 3.1

3.1 A hash table has $m = 11$ buckets with chaining, and the keys 6, 17, 28, 39, 50 arrive in order.

Compute each key's bucket, and give every non-empty chain after all insertions. State α , the expected successful-search cost $1 + \alpha/2$, and the exact worst-case comparison count for searching 50 in your table. [8]

3.1

- Buckets: $6 \bmod 11 = 6$; $17 \bmod 11 = 6$; $28 \bmod 11 = 6$; $39 \bmod 11 = 6$; $50 \bmod 11 = 6$ ✓.
- One chain: $6 \rightarrow 17 \rightarrow 28 \rightarrow 39 \rightarrow 50$ in bucket 6 ✓.
- $\alpha = 5/11 \approx 0.45$; expected successful cost $1 + 0.45/2 \approx 1.23$ comparisons ✓.
- Worst case for 50: five comparisons – the whole chain ✓.

Exercise 3.2

3.2 The same five keys enter an open-addressed table of 11 slots with linear probing.

Give the slot each key lands in and the probe count for each insertion, with the running total. State the total had the keys hashed to five distinct buckets. Name the effect that makes the first total larger, and the load-factor policy that controls it. [7]

Exercise 3.2

3.2 The same five keys enter an open-addressed table of 11 slots with linear probing.

Give the slot each key lands in and the probe count for each insertion, with the running total. State the total had the keys hashed to five distinct buckets. Name the effect that makes the first total larger, and the load-factor policy that controls it. [7]

3.2

- Slots: $6 \rightarrow 6$; 17 probes $6, \rightarrow 7$; $28 \rightarrow 6, 7, \rightarrow 8$; $39 \rightarrow 6, 7, 8, \rightarrow 9$; $50 \rightarrow 6, 7, 8, 9, \rightarrow 10$ ✓.
- Probe counts 1, 2, 3, 4, 5 – running total 15 ✓.
- Five distinct buckets would cost exactly 5 probes total ✓.
- The effect is clustering: the occupied run grows the probe path ✓.
- Policy: keep α under about 0.75 and resize ✓.