Data Structures and Algorithms

Lecture 19

Aniket Basu Roy

2026-03-06 Fri

Contents

1 Agenda

1.1 Recap: Hashing with Chaining

1.2 Open Addressing

1.3 Probing Strategies

2 Recap: Hashing with Chaining

- Hash table of size m; keys from universe U.
- Hash function $h : U \to \{0, \ldots, m-1\}$.
- Collisions resolved by linked lists (chains) at each slot.
- Under SUHA, expected search time: $\Theta(1+\alpha)$ where $\alpha = n/m$.
- Universal hashing: choose h randomly from a universal family to guarantee $O(1+\alpha)$ expected regardless of input.

3 Open Addressing

3.1 Idea

Store **all** elements directly in the table — no external lists. On collision, **probe** alternative slots according to a **probe sequence**.

3.2 Probe Sequence

For key k, the probe sequence is a permutation of $\{0, 1, \ldots, m-1\}$:

$$h(k,0),\ h(k,1),\ \ldots,\ h(k,m-1)$$

We try slots in this order until we find an empty slot.

3.3 Constraint

The load factor must satisfy $\alpha < 1$ (at most $m-1$ elements).

4 Open Addressing: Operations

```
HASH-INSERT(T, k)
1.  i = 0
2.  repeat
3.      j = h(k, i)
4.      if T[j] == NIL
5.          T[j] = k
6.          return j
7.      else i = i + 1
8.  until i == m
9.  error "hash table overflow"

HASH-SEARCH(T, k)
1.  i = 0
2.  repeat
3.      j = h(k, i)
4.      if T[j] == k
5.          return j
6.      i = i + 1
7.  until T[j] == NIL or i == m
8.  return NIL
```

4.1 Deletion

Deletion is tricky: cannot simply set $T[j] = \text{NIL}$ (breaks search chains). Use a special marker DELETED; treat as occupied for INSERT, empty for SEARCH.

5 Linear Probing

$$h(k, i) = (h'(k) + i) \bmod m$$

where h' is an ordinary hash function.

- The probe sequence wraps around the table.
- Simple and cache-friendly.

5.1 Problem: Primary Clustering

Long runs of occupied slots build up. A key hashing into the middle of a cluster must probe through the whole cluster. This makes future insertions and searches slower.

```
Slots: [_, _, A, B, C, D, _, _, E, _]
New key hashes to slot 2 → probes 2,3,4,5,6 (cluster grows)
```

6 Quadratic Probing

$$h(k, i) = (h'(k) + c_1 i + c_2 i^2) \bmod m$$

for constants c_1, c_2 with $c_2 \neq 0$.

- Spreads probes more broadly than linear probing.
- Choice of c_1, c_2, m must ensure all m slots are visited (e.g., m a prime and $c_1 = c_2 = 1/2$ works).

6.1 Problem: Secondary Clustering

If $h'(k_1) = h'(k_2)$, keys k_1 and k_2 follow the **same** probe sequence $\Rightarrow$ they always collide with each other again.