BCE202L DS&A

Hashing

What is hashing?

- Linear search → $O(n)$
- Binary search → $O(\log n)$ plus time to sort
- Storing, retrieving, and Searching data in $O(1)$ time
- Hash Table → The data structure used
- Search keys→ Data stored in hash table
- Store elements at the index; the simplest hash function
- But space restrictions

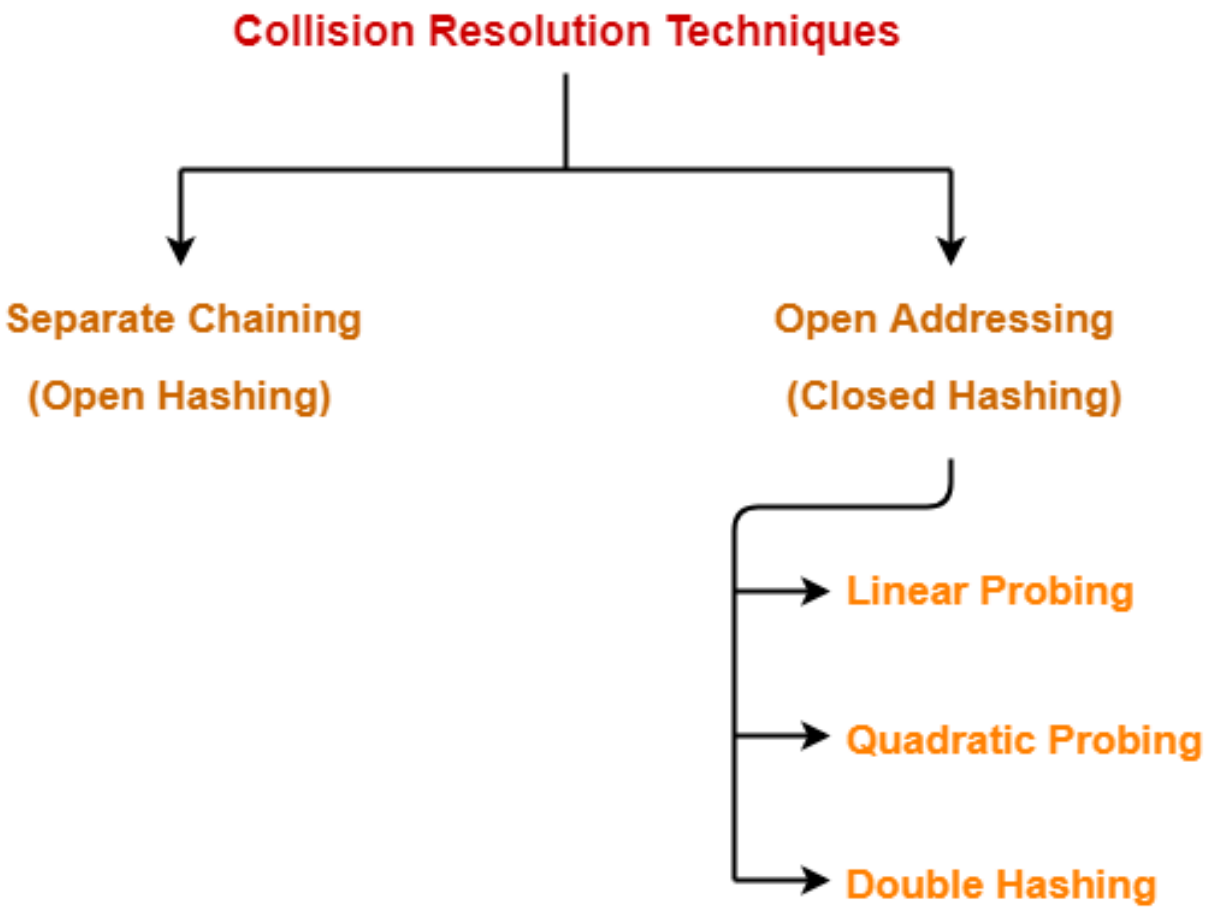
Hash function

- The simplest and most popular hash function is:$h(k) = k$
- To reduce space requirements use division method → Take mod size
- If $h(k)$ is defined, Location, $l(k) = h(k) \bmod size$
- Other methods:
- Mid square method, folding method

Collision Demo

- Use division method
- Difficulty: Collision
- Collision resolution is essential
- Demo $h(k) = k$ store 8,3,13,68,23,14,17,53 with $size = 10$
- Demo with $h(k) = 2k + 3$ store 3,2,9,6,11,13,7,12 with $size = 10$

Collision Resolution



Credit: www.gatevidyalay.com

Open Hashing (Separate Chaining)

- aka Closed Addressing/Chaining
- In open hashing, keys are stored in linked lists attached to cells of a hash table
- Demo → Do examples in the previous slide
- Advantages:- Deletion is easy (rearrange pointers)
- Disadvantages:- Searching/deletion takes $O(n)$ time (in worst case), Use of extra space even if space is available in hash table
- Load factor (λ) = #keys/#buckets

Closed Hashing (Open Addressing)

- Three Techniques
  1. Linear Probing
  2. Quadratic Probing
  3. Double Hashing

Linear Probing

- The principle: In case of a collision insert the key in the next available location
- Location, $l(k) = h(k) \bmod size$, in case of collision
- New location $l'(k) = (l(k) + i) \bmod size$ where $i = 1,2, \dots, size - 1$
- Demo with $h(k) = 2k + 3$ store 3,2,9,6,11,13,7,12 with $size = 10$. Use division method

Liner Probing

- Advantages:- No extra space
- Disadvantages:-
- Searching takes $O(n)$ time (in worst case)
- Deletion stops search
- Primary clustering

Quadratic Probing

- The principle
- In case of a collision at location l, insert k at the next free location $(l + i^2) mod\ n$ where $i = 0$ to $n - 1$
- Demo-1:- Previous example
- Demo-2:- 42,16,91,33,18,27,36,62 $h(k) = k$, size=10, division method

Quadratic Probing

- Advantages:- No extra space, primary clustering resolved
- Disadvantages:-
  - Search can take $O(n)$ time
  - Secondary clustering (two elements following the same probe sequence)
  - No guarantee of finding a free slot, despite free slots

Double Hashing

- Use two hash functions
- $h_1\,(k)$ and $h_2(k)$
- First compute $l_1(k)$ using h_1 and upon collision at that location, insert k at the next free location such that

$(l_1(k) + il_2\,(k))mod\,n$ where $i\ = 1$ to $n - 1$, where $l_2(k)$ is the location identified using h_2

- Keep n prime so that result of $h_2(k)$ and n will be relatively prime to have a uniform distribution of keys

Double Hashing

- Demo:
- Keys: 20,34,45,70,56; size = 11; $h_1(k) = k$; $h_2(k) = 8 - (k \bmod 8)$
- Exercise:
- Keys: 3,2,9,6,11,13,7,12; size=10; $h_1(k) = 2k + 3$; $h_2(k) = 3k + 1$ (Can 13 be inserted? Why(not)?)

Double Hashing

- Advantages:-
  - No extra space
  - No primary clustering
  - No secondary clustering
- Disadvantages:-
  - Searching → $O(n)$

Rehashing

- Load factor (λ) = #keys(K)/#buckets(N)
- $\lambda < 1$ is ideal.
- What if $\lambda \geq 1$, increase N to a larger value N', where N' = the closest prime number to $2N$
- This is rehasing
  - Increment buckets to N' (usually the smallest prime number > 2N)
  - Modify hash function k mod N to k mod N'
  - Apply $h'(k)$ to the existing keys

Thank you..