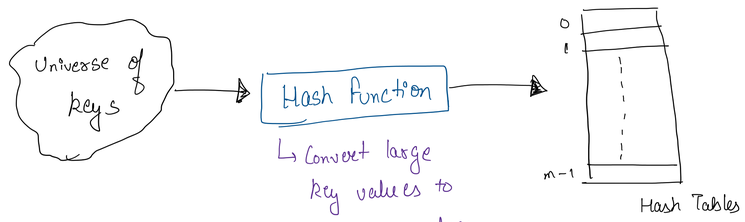
Day 11

Friday, 17 March 2023 9:08 PM

HASHING



→ Used in dictionaries
→ Where we have set of keys
→ Search, Insert, Delete ⟩ O(1) on average
→ Data is not sorted
→ Not useful for:
- finding closest value] AVL or RB Tree
- Sorted data
- Prefix Searching] Trie

↳ Convert large key values to smaller values
↳ Every time we map key, it should generate same index
↳ take O(1) or O(len(s)) time. ↳ strings

→ Collision Handling
↳ Chaining
↳ Open Addressing
- Linear Probing
- Quadratic Probing
- Double Hashing

unordered_set → Based on hashing Unlike set (RB tree)

- unordered_set<int> s; (int: Data Type, s: Variable Name)
- Does not maintain order of elements
- → begin() : iterator to starting of set
  → end() : iterator beyond the end of set
  → insert() → find()
  → size() → count() → either 1 or 0
  → clear() → erase()

unordered_map → Based on hashing Unlike map (RB tree)

- unordered_map<int, int> mp; (first int: datatype of key, second int: Data type of value, mp: Map name)
- Store key-value pairs.
- Does not maintain order of elements
- → begin() : iterator to starting
  → end() : iterator beyond the end
  → insert() → find()
  → size() → count()
  → clear() → erase()

Problem:1 Count Distinct Elements

I/P: {15, 14, 12, 12, 13, 3, 15}
O/P: 5

Problem:2 Count frequencies of element within array

I/P: {10, 10, 12}
O/P: 10 2
12 1

Problem:3 Intersection of Two arrays (leetcode 349)

Problem:4 Find if there is a pair with given sum in the array

Problem:5 Finding Pairs with Certain Sum (leetcode 1865)