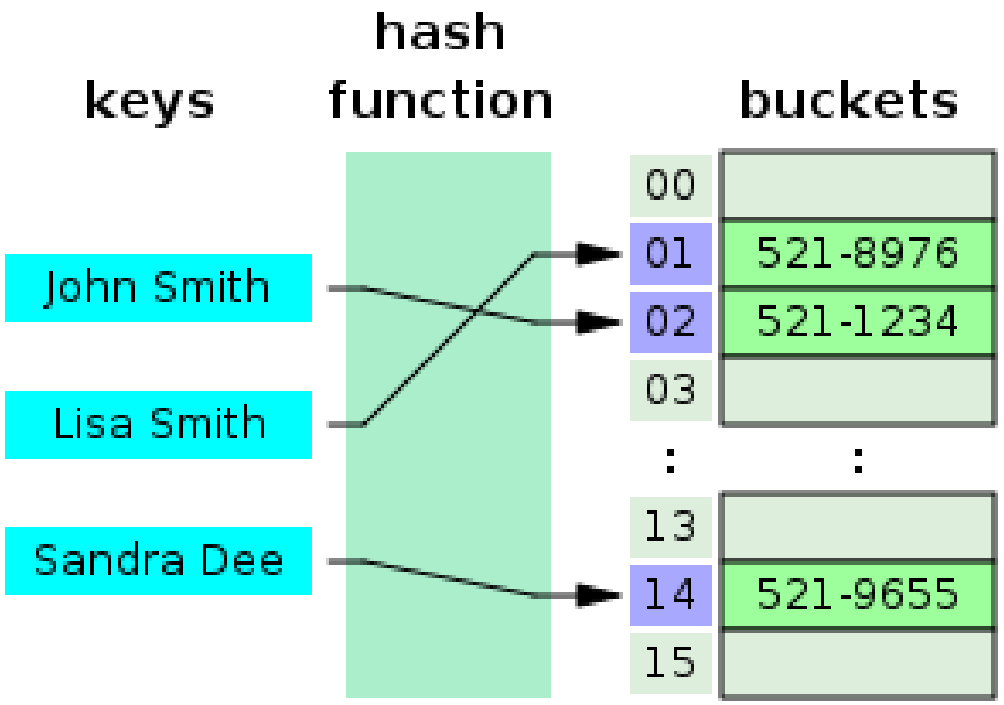
Miscellaneous

Hash Table

- Hash Table is a data structure which stores data in an associative manner. In a hash table, data is stored in an array format, where each data value has its own unique index value.
- Access of data becomes very fast if we know the index of the desired data.

Hash Table



- Insert (pair)
- index = HashFunction (pair.key)
- IF array[index] is empty
- array[index] = pair
- ELSE
- HandleCollision (index, pair)
- ENDIF
- End Insert

https://www.journaldev.com/35238/hash-table-in-c-plus-plus#:~:text=A%20Hash%20Table%20in%20C,value%20at%20the%20appropriate%20location.

Hash Table

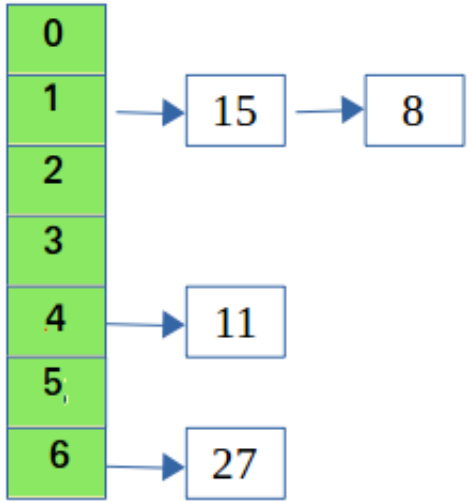
- Any Hash Table implementation has the following three components:
  - A good Hash function to map keys to values
  - A Hash Table Data Structure that supports insert, search and delete operations.
  - A Data Structure to account for collision of keys

Managing Collision

- Chain hashing avoids collision. The idea is to make each cell of hash table point to a linked list of records that have same hash function value

Let's say hash table with 7 buckets (0, 1, 2, 3, 4, 5, 6)

Keys arrive in the Order (15, 11 , 27 , 8)



https://www.geeksforgeeks.org/c-program-hashing-chaining/

Priority Queue

- https://www.cplusplus.com/reference/queue/priority_queue/
- Priority queues are a type of container adapters, specifically designed such that the first element of the queue is the greatest of all elements in the queue and elements are in nonincreasing order (hence we can see that each element of the queue has a priority {fixed order})
- Sample code:
- https://www.geeksforgeeks.org/priority-queue-in-cpp-stl/