

# CS2023 - Data Structures and Algorithms Take Home Assignment

Submission by: Sajeev Kugarajah (210554M)

## Questions

1. What is Hash Table?

hash table is a data structure which stores data in an array with a unique key for each data as key-value pairs. it allows efficient insertion, deletion and lookup for key-value pairs. hash tables provide fast access to the data. the time complexity for all above operations are constant  $O(1)$  in a hash table.

2. Define what is a Hash Function?

hash function is a function which creates unique keys for datas. It takes input data and returns unique fixed-size output key which referred as hash code. hash function distributes the keys evenly across the data array as indices.

3. What is the significance of load factor of a Hash Table?

load factor of a hash table is the ratio of the filled slots in a hash table and available slots. it defines how full the hash table is. the load factor is very significant as it affects the performance of a hash table. probability of collision occurrence increases with the load factor. so to reduce collisions and maintain a good performance of a hash table it's necessary to maintain a correct load factor.

4. Explain some techniques to handle collision in Hash Table

Separate chaining :-

Chaining is a technique where each key-value pairs of a hash table is stored as a linked list. when collision occurs with same hash codes, simply a new lined list created and appended to the existing array. chaining is effective technique but the performance will be affected as the length of the linked list grows.

Open addressing :-

in this method when a collision occurs, the hash table looks for an alternative slot to store the key-value pair. this can be achieved through different probing methods such as linear probing, quadratic probing and double hashing.

Linear probing :  $h(k, i) = (h'(k) + i) \bmod m$  (looks for the next available slot)

Quadratic probing :  $h(k, i) = (h'(k) + c_1*i + c_2*i^2) \bmod m$  (uses a function generate slot)

Double hashing :  $h(k, i) = (h_1(k) + i*h_2(k)) \bmod m$  (uses 2 functions to choose next slot)