1 Introduction and Objective

In this tutorial, we will continue our discussion about Non-Linear Data Structures. Next up is Hash Table, one possible implementation of Table ADT (unordered). We will heavily use https://visualgo.net/en/hashtable in this tutorial.

2 Tutorial 06 Questions

Hash Table Basics

Q1). A good hash function is essential for good Hash Table performance. A good hash function is easy/efficient to compute and will evenly distribute the possible keys. Comment on the flaw (if any) of the following hash functions. Assume that the load factor \( \alpha = \frac{\text{number of keys}}{\text{Hash Table size}} \) = 0.3 for all the following cases:

1. \( M = 100 \). The keys are positive even integers. The hash function is \( h(key) = key \mod 100 \).

2. \( M = 1009 \). The keys are valid email addresses. The hash function is \( h(key) = (\text{sum of ASCII values of each of the last 10 characters}) \mod 1009 \). See http://www.asciitable.com for ASCII values.

3. \( M = 101 \). The keys are integers in the range of \([0, 1000]\). The hash function is \( h(key) = key \ast \text{random} \mod 101 \), where \( 0.0 \leq \text{random} \leq 1.0 \).

Q2). Hashing or No Hashing: Hash Table is a Table ADT that allows for \textbf{search(v)}, \textbf{insert(new-v)}, and \textbf{delete(old-v)} operations in O(1) average-case time, if properly designed. However, it is not without its limitations. For each of the cases described below, state if Hash Table can be used. If not
possible to use Hash Table, explain why is Hash Table not suitable for that particular case. If it is possible to use Hash Table, describe its design, including:

1. The <Key, Value> pair
2. Hashing function/algorithim
3. Collision resolution (including second hash function/algorithim for double hashing)

The cases are:

1. A mini-population census is to be conducted on every person in your (not so large) neighborhood. We are only interested in storing every person’s name and age. The operations to perform are: retrieve age by name and retrieve name(s) by age.

2. A much larger population census is also conducted across the country, similarly containing only every person’s name and age. The operation to perform is: retrieve names of people eligible for voting. Only people above legal age 17 years old (or older) are eligible for voting. However, we still need to store the rest of the data.

3. A different population census similarly contains only the name and age of every person. The operation to perform is: retrieve people with a given last name.

4. In a scientific experiment, data on energy converted for different runs is collected. Each run has a different velocity and distance, both floating point numbers. The operation to perform is: retrieve the energy value for a given velocity and distance. [Hint: What would happen when you hash a floating-point number?]

5. A grades management program stores a student’s index number and his/her final marks in one GCE ‘O’ Level subject. There are 1,000,000 students, each scoring final marks in [0.0, 100.0]. The operation to perform is: retrieve a list of students who passed in ranking order (highest final marks to passing marks). A student passes if the final marks are more than 65.5. Whether a student passes or not, we still need to store all students’ performance.

**Basic Hash Table Stuffs**

Q3). Quick check: Let’s review all 4 modes of Hash Table (use the Exploration mode of [http://visualgo.net/en/hashtable](http://visualgo.net/en/hashtable)). During the tutorial session, the tutor will randomize the Hash Table size $M$, the selected mode (LP, QP, DH, or SC), the initial keys inside, and then ask student to $\text{Insert(random-integer)}$, $\text{Remove(existing-integer)}$, or $\text{Search(integer)}$ operations.

**Hash Table Discussions**

Q4). The following topics require deeper understanding of Hash Table concept. Please review [https://visualgo.net/en/hashtable?slide=1](https://visualgo.net/en/hashtable?slide=1) use the Exploration Mode, or Google around to help you find the initial answers and we will discuss the details in class.
1. What is/are the main difference(s) between List ADT basic operations (see https://visualgo.net/en/list?slide=2-1) versus Table ADT basic operations (see https://visualgo.net/en/hashtable?slide=2-1)?

2. At https://visualgo.net/en/hashtable?slide=4-4, Steven mentions about Perfect Hash Function. Now let’s try a mini exercise. Given the following strings, which are the names of Steven’s current family members: {“Steven Halim”, “Grace Suryani Halim”, “Jane Angelina Halim”, “Joshua Ben Halim”, “Jemimah Charissa Halim”}, design any valid minimal perfect hash function to map these 5 names into index [0..4] without any collision. Steven and Grace are not planning to increase their family size further so you can assume that \( N = 5 \) will not change.

3. Thus far, which collision resolution technique is better (in your opinion or Google around): One of the Open Addressing technique (LP, QP, DH) or the Separate Chaining technique?

4. Which non-linear data structure should you use if you have to support the following three operations: 1). many insertions, 2) many deletions, and 3) many requests for the data in sorted order?