

CS2040C Semester 1 2020/2021  
Data Structures and Algorithms

## Tutorial 05 - Midterm Quiz Review; Hash Function

For Week 07

Document is last modified on: September 28, 2020

### 1 Introduction and Objective

Welcome back from recess week =). I hope that you are (a bit) fresher now.

In the early part of the tutorial component of this session, we will discuss the solutions and a few common mistakes that were found during (early) grading. Steven will upload the updated midterm quiz modal answer file with full statistics at unspecified time later.

Then, we will discuss a bit of <https://visualgo.net/en/hashtable> in this tutorial that were only covered as e-Lecture recordings on Week 06.

### 2 Tutorial 05 Questions

#### Midterm Quiz Solutions + Review

Q1). Brace for impact...

#### Hash Function Basics

Q2). Which of the following is the best (string) hash function?

1. `int index = (rand() * (key[0] - 'A')) % N;`
2. `int index = (key[0] - 'A') % N;`
3. `int index = hash_function(key) % N;`

where

- `rand()` is a function that returns a pseudo-random integral number in the range between 0 and `RAND_MAX` (This value is library-dependent, but is guaranteed to be at least 32767 on any standard library implementation).

- `key` is a C++ string
- $N$  is the hash table size, usually a prime number
- `hash_function(v)` is as shown in <https://visualgo.net/en/hashtable?slide=4-7>

Q3). 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 (integer) hash functions. Assume that for this question, the load factor  $\alpha = \text{number of keys } N / \text{Hash Table size } M = 0.3$  (i.e., low enough) for all cases below:

1.  $M = 100$ . The keys are positive even integers. The hash function is  $h(\text{key}) = \text{key} \% 100$ .
2.  $M = 100$ . The keys are non-negative integers in the range of  $[0, 10\,000]$ . The hash function is  $h(\text{key}) = \text{floor}(\text{sqrt}(\text{key})) \% 100$ .
3.  $M = 101$ . The keys are integers in the range of  $[0, 1\,000]$ . The hash function is  $h(\text{key}) = \text{floor}(\text{key} * \text{random}) \% 101$ , where  $0.0 \leq \text{random} \leq 1.0$ .

## Hands-on 5

TA will run the second half of this session with a few to do list:

- Very quick review of C++ STL `std::unordered_set` and `std::unordered_map`,
- Do a(nother) sample speed run of VisuAlgo online quiz that are applicable so far, e.g., <https://visualgo.net/training?diff=Hard&n=5&tl=5&module=hashtable>.
- PS2 Debrief,
- Then, live solve another chosen Kattis problem involving (Hash) Table ADT.

## Problem Set 3

We will end the tutorial with **short algorithmic** discussion of PS3.

As we still have Week 08 before PS3 is due, then TAs are not supposed to reveal the algorithmic ideas of 100+100 solutions publicly (yet).