Data Structure I: Tutorial 6

Exercise 1. *Describe how data are stored using hash tables. Compare hash tables with sorted array and search trees.*

Basic terms

- Universe $U = \{0, 1, \dots, u-1\}$ of all elements
- Represent a subset $S \subseteq U$ of size n
- Supported operations are Find, Insert and Delete an element of U
- Store S in an array of size m using a hash function $h : U \to M$ where $M = \{0, 1, \dots, m-1\}$
- Collision of two elements $x, y \in S$ means $h(x) = h(y)$

Exercise 2.

- *For a given set of elements S, is it possible to construct a hash function without collisions on S?*
- *Is it possible to construct a hash function which has no collision for every set of n elements S?*
- *For a given hash function $f : U \to M$, is it possible to find a set of n elements which are all hashed to the same position?*

Exercise 3. *Consider a simple hash function $h(x) = x \mod m$. When this hash function is sufficient and useful?*

Exercise 4. *Consider that we need to store some data for every edge of a directed graph. We can use a hash table which for every pair vertices (integers) assign the data. C++ provides functions for hashing integers, string, etc. but there is no hash function for pairs or tuples. However, we can find many websites which provides solutions similar to the following one.*

```
template <class T1, class T2>
size_t operator()(const pair<T1, T2>& p) const {
    return hash<T1>{}(p.first) ^ hash<T2>{}(p.second);
}
```

- *https://www.geeksforgeeks.org/how-to-create-an-unordered_map-of-pairs-in-c/*
- *https://stackoverflow.com/questions/72637402*
- *https://stackoverflow.com/questions/32685540*

Now, consider that our graph is an oriented path from 1 to n with edges from i to $i+1$ for $i = 1, \dots, n-1$. In which positions of the hash table these edges are stored?

Note that the hash function for integers just return the argument:

```
size_t operator()(size_t p) const {
    return p;
}
```

Definition 5. *A system $\mathcal{H}$ of hashing functions is c-universal, if for every $x, y \in U$ with $x \neq y$ the number of functions $h \in \mathcal{H}$ satisfying $h(x) = h(y)$ is at most $\frac{c|\mathcal{H}|}{m}$ where $c \geq 1$. Equivalently, a system $\mathcal{H}$ of hashing functions is c-universal, if uniformly chosen $h \in \mathcal{H}$ satisfies $P[h(x) = h(y)] \leq \frac{c}{m}$ for every $x, y \in U$ with $x \neq y$.*

Exercise 6. *When do we need more than one hash function?*

Common Vulnerabilities and Exposures:

- *PHP: CVE-2011-4885*
- *Ruby: CVE-2011-4815*
- *Apache Geronimo: CVE-2011-5034*

Definition 7. *Let $p \geq |U| \geq m$ be a prime where $U = [u]$. We define the hash function*

$$h_{a,b}(x) = (ax + b \mod p) \mod m.$$

Hashing system Multiply-mod-prime is

$$\mathcal{H} = \{h_{a,b};\, a, b \in [p]\}$$

Theorem 8. *Hash system Multiply-mod-prime is 2-universal.*

Definition 9. *Assume that $u = 2^w$ and $m = 2^l$ for some integer w, l. We define the hash function*

$$h_a(x) = (ax \mod 2^w) >> (w - l)$$

Hashing system Multiply-shift is

$$\mathcal{H} = \{h_a;\, a \textit{ is odd } w\textit{-bits integer } \}$$

Theorem 10. *Hash system Multiply-shift is 1-universal.*