# CSE, BUET

# Bangladesh University of Engineering & Technology

## 1805004

## COURSE
### CSE 208

### Data Structures and Algorithms II Sessional

# Report On:

### Performance of various hashing techniques for collision resolution

# Topic:

### Hash Table

Author:
Syed Jarullah Hisham
**Roll: 1805004**
CSE'18 Section A1

January 18, 2022

# Used Hash Functions

We have used two different hash functions and an auxiliary hash functions to make the report of performance for three different collision resolution techniques. The functions are depicted below:

### Hash Function 1:

```java
public int HashFunc1(String str) {
    int hashVal = MODULUS;
    int higherOrder;

    for (int i = 0; i < str.length(); i++) {
        higherOrder = hashVal & 0xFE000000;
        hashVal ^= (hashVal << MAGIC_PRIME_2) ^ (hashVal >> (Integer.SIZE - MAGIC_PRIME_2)) ^
            str.charAt(i);
        hashVal ^= (higherOrder << (Integer.SIZE - MAGIC_PRIME_2));
        hashVal += MAGIC_PRIME;
    }

    return (hashVal % tableLength + tableLength) % tableLength;
}
```

### Hash Function 2:

```java
// CRC 32 hashing
public int HashFunc2(String str) {
    long crc32 = 0xffffffff;

    for (int i = 0; i < str.length(); i++) {
        crc32 = (crc32 >> 8) ^ this.crc32_table[(int) ((crc32 & 0xff) ^ str.charAt(i))];
    }

    crc32 ^= 0xffffffff;
    return (int) (crc32 % tableLength);
}
```

### Auxiliary Hash Function:

```java
public int auxiliaryHashFunc(String str) {
    int hashVal = MAGIC_PRIME_2;

    for (int i = 0; i < str.length(); i++) {
        hashVal = (MAGIC_PRIME * hashVal + str.charAt(i)) % tableLength;
    }

    return (hashVal % tableLength + tableLength) % tableLength;
}
```

# Performance Report

## Table 1: For input size 10000 and search 1000

| Collision Resolution Method | Hash 1 | | Hash 2 | |
|---|---|---|---|---|
| | *No. of collisions* | *Avg. Probes* | *No. of collisions* | *Avg. Probes* |
| **Separate Chaining Method** | 3673 | *1.53* | *3712* | *1.487* |
| **Double Hashing Method** | *60137* | *6.586* | *59996* | *7.321* |
| **Custom Probing Method** | *61131* | *7.538* | *63359* | *7.151* |

## Table 2: For input size 50000 and search 5000

| Collision Resolution Method | Hash 1 | | Hash 2 | |
|---|---|---|---|---|
| | *No. of collisions* | *Avg. Probes* | *No. of collisions* | *Avg. Probes* |
| **Separate Chaining Method** | 18442 | *1.50* | *18466* | *1.5036* |
| **Double Hashing Method** | *336651* | *7.54* | *334761* | *8.0* |
| **Custom Probing Method** | *343456* | *7.268* | *329762* | *8.166* |

## Table 3: For input size 100000 and search 10000

| Collision Resolution Method | Hash 1 | | Hash 2 | |
|---|---|---|---|---|
| | *No. of collisions* | *Avg. Probes* | *No. of collisions* | *Avg. Probes* |
| **Separate Chaining Method** | 36716 | *1.502* | *36810* | *1.497* |
| **Double Hashing Method** | *727672* | *8.007* | *747611* | *8.592* |
| **Custom Probing Method** | *750367* | *7.96* | *755167* | *8.284* |