

Student ID: 1805064

Table 1: Performance of various techniques for collision resolution with two different hash functions

For Hash Table size = 20007

	Hash1		Hash2	
	Number of collisions	Average probes	Number of collisions	Average probes
Chaining Method	2115	1.22	2305	1.25
Double Hashing	3729	1.33	4055	1.364
Custom Probing	3900	1.364	4206	1.394

For Hash Table size = 30007

	Hash1		Hash2	
	Number of collisions	Average probes	Number of collisions	Average probes
Chaining Method	1510	1.15	1494	1.15
Double Hashing	2173	1.194	2099	1.196
Custom Probing	2173	1.197	2134	1.19

For Hash Table size = 40007

	Hash1		Hash2	
	Number of collisions	Average probes	Number of collisions	Average probes
Chaining Method	1150	1.1	1160	1.12
Double Hashing	1455	1.12	1522	1.125
Custom Probing	1536	1.148	1480	1.11

```
///HashOne Function
int HashTable::hash1(string key)
{
    long long int index= 7;
    char ch[key.length()];
    for(int i=0; i<key.length(); i++)
    {
        ch[i]= key[i];
        index= index+ index*13 + ch[i];
    }
    return index%capacity;
}

///HashTwo Function
int HashTable::hash2(string key)
{
    long long int sum=0;
    char ch[key.length()];
    for (int i = 0; i < key.length(); i++)
    {
        ch[i]= key[i];
        sum= sum+ ch[i]<< (3*i);
        sum= sum%capacity;
    }
    return sum% capacity;
}
```

```
///auxHash Function
int HashTable::auxHash(string key)
{
    long long int index= 7;
    char ch[key.length()];
    for(int i=0; i<key.length(); i++)
    {
        ch[i]= key[i];
        index=  index+ index*7 + ch[i]*5;
    }
    return (capacity-13) - index%(capacity-13);
}
```