NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

SC1007 Data Structures and Algorithms

Lab 7: Hash Tables

Q1 Implement a closed addressing hash table to perform insertion and key searching. The insertion may not have to insert at the end of the link-list. The function prototype is given below:

```
def hash_search(self, key):
def hash_insert(self, key):
```

The default load factor is 3. The number of hash slots of the created hash table depends on the provided amount of data.

Q2 Implement an open addressing hash table with linear probing to perform insertion, deletion, and key searching. The function prototype is given below:

```
def hash_search(self, key):
def hash_insert(self, key):
def hash_delete(self, key):
```

You need to specify the number of hash slots at the beginning. Each slot has two fields, one for key, and the other to indicate whether the key is deleted. If a key is deleted, it will be marked as deleted and it can be replaced by other keys. The hash function H(k,i) =(H'(k) + i) mod h, where H'(k) = k mod h.