

# ECE374 Fall2020

## Lab7: Hash Table Insertion

Name: Zhang Yichi 3180111309

Nov. 16<sup>th</sup> 2020

### 1 Introduction

In this lab, we will implement and analyze the hash algorithm with quadratic probing and double hashing.

### 2 Python Code for Hash

```
def quadratic_probing_insert(T,k):
    i = 0
    while (i < 21):
        j = (k+i+3*i**2) % 21
        if T[j] == None:
            T[j] = k
            return j
        else:
            i = i + 1

def double_hashing_insert(T,k):
    i = 0
    while (i < 21):
        j = (k + i*(1 + (k % 20))) % 21
        if T[j] == None:
            T[j] = k
            return j
        else:
            i = i + 1
```

Each function implements a hash algorithm. We take  $h'(k) = k$ ,  $m = 21$ ,  $c_1 = 1$ ,  $c_2 = 3$ ,  $h_1(k) = k$ ,  $h_2(k) = 1 + (k \bmod (m - 1))$ . The first one is hash with quadratic probing  $h(k, i) = (h'(k) + c_1i + c_2i^2) \bmod m = (k + i + 3i^2) \bmod 11$  and the second one is hash with double hashing  $h(k, i) = (h_1(k) + ih_2(k)) \bmod m = (k + i(1 + (k \bmod 20))) \bmod 21$ .

### 3 Test Example for Hash

```

m = 21
quadratic = [None]*m
double = [None]*m
A = [7, 22, 44, 43, 27, 89, 30, 64, 85]
for i in A:
    quadratic_probing_insert(quadratic,i)
    double_hashing_insert(double,i)

print("hash insertion with quadratic probing")
print(quadratic)
print("hash insertion with double hashing")
print(double)

```

```

hash insertion with quadratic probing
[None, 22, 44, None, None, 43, 27, 7, None, 89, 85, None, None, 30, None, 64, None, None, None, None]
hash insertion with double hashing
[None, 22, 44, None, None, 43, 27, 7, None, 30, None, 64, None, 85, None, 89, None, None, None, None]

```

Figure 1: hash insert test result

## 4 Time Complexity Analysis

Both the insertion with quadratic probing and double hashing have the time complexity of  $\Theta(1+\alpha)$  where  $\alpha = \frac{n}{m}$ .