

ECE374 Fall2020

Lab8: Skip List

Name: Zhang Yichi 3180111309

Nov. 30th 2020

1 Introduction

In this lab, we will implement and test some functions of the skip list.

2 Python Code for Skip List

```
import random

class Node:
    def __init__(self, key, value):
        self.key = key
        self.value = value
        self.next = []

class SkipList:
    def __init__(self, elements, MaxLevel=5):
        self.MaxLevel = MaxLevel
        self.header = Node(-float('inf'), None)
        self.header.next = [None]*self.MaxLevel
        self.level = 1
        for x in elements:
            self.insert(x[0], x[1])

    def search(self, key):
        x = self.header
        for i in range(self.level-1, -1, -1):
            while x.next[i] and x.next[i].key < key:
                x = x.next[i]
        x = x.next[0]
        if x and x.key == key:
            return x.value
        else:
            return None

    def insert(self, key, value):
        lvl = self.randomlevel()
```

```

newnode = Node(key,value)
newnode.next = [None]*lvl
update = [self.header]*lvl
x = self.header
for i in range(lvl-1,-1,-1):
    while x.next[i] and x.next[i].key < key:
        x = x.next[i]
    update[i] = x
x = x.next[0]
if x and x.key == key:
    x.value = value
    return
if self.level < lvl:
    self.level = lvl
for i in range(lvl):
    newnode.next[i] = update[i].next[i]
    update[i].next[i] = newnode

def delete(self,key):
    update = [None]*self.level
    x = self.header
    for i in range(self.level-1,-1,-1):
        while x.next[i] and x.next[i].key < key:
            x = x.next[i]
        update[i] = x
    x = x.next[0]
    if x and x.key == key:
        for i in range(self.level):
            if update[i].next[i] and update[i].next[i].key == key:
                update[i].next[i] = x.next[i]
        while self.level > 1 and self.header.next[self.level-1]==None:
            self.level -= 1

def randomlevel(self):
    lvl = 1
    while random.random() < 0.5 and lvl < self.MaxLevel:
        lvl += 1
    return lvl

def printSL(self):
    for i in range(self.level-1,-1,-1):
        x = self.header
        print("header->",end="")
        while x.next[i]:
            print((x.next[i].key,x.next[i].value),end="->")
            x = x.next[i]
        print("tail\n")

def printKey(self):
    x = self.header
    while x.next[0]:
        x = x.next[0]
    print(f"Key: {x.key}, Number of Levels: {len(x.next)}")
    print()

```

Skip list is a data structure that stores the elements in a bunch of linked lists. The linking is decided randomly which provide a shortcut for searching. The core functions of the skip list is insert, search and delete. To search an element, we start from the top level with the shortest path and then go down. To insert an element, we first find its place using the similar routine of searching and then insert it or replace it when the key is already there. If we insert a new node, we have to update its linking by first determine its hight and then connect it with the previous node and the next node corresponding to the hight. Deletion is similar to the insert. We first find the element, delete it and update the linking.

3 Test Example for Skip List

```
# build the Skip List and test insert
inputlist = [[random.randint(1,10) for j in range(1,3)] for i in range(1,11)]
print(f"the input list is {inputlist}\n")
SL = SkipList(inputlist)
SL.printSL()
SL.printKey()

# Test search
x = random.choice(inputlist)[0]
print(f"test search: the value of {x} is {SL.search(x)}")
print()

# Test delete
SL.delete(random.choice(inputlist)[0])
SL.delete(random.choice(inputlist)[0])
SL.printSL()
SL.printKey()
```

```
the input list is [[9, 6], [2, 3], [3, 5], [3, 6], [1, 10], [7, 8], [1, 8], [4, 2], [9, 4], [5, 5]]
header->(4, 2)->tail
header->(2, 3)->(3, 6)->(4, 2)->tail
header->(1, 8)->(2, 3)->(3, 6)->(4, 2)->(5, 5)->(7, 8)->(9, 4)->tail
Key: 1, Number of Levels: 1
Key: 2, Number of Levels: 2
Key: 3, Number of Levels: 2
Key: 4, Number of Levels: 3
Key: 5, Number of Levels: 1
Key: 7, Number of Levels: 1
Key: 9, Number of Levels: 1
```

Figure 1: Skip list test result.

```
test search: the value of 5 is 5
```

Figure 2: Skip list search test result.

```
header->(2, 3)->tail  
  
header->(1, 8)->(2, 3)->(5, 5)->(7, 8)->(9, 4)->tail  
  
Key: 1, Number of Levels: 1  
Key: 2, Number of Levels: 2  
Key: 5, Number of Levels: 1  
Key: 7, Number of Levels: 1  
Key: 9, Number of Levels: 1
```

Figure 3: Skip list deletion test result.

Here we implement and test the insert, search and delete functions of the skip list.

4 Time Complexity Analysis

Average:

Insert: $O(\log n)$

Search: $O(\log n)$

Delete: $O(\log n)$