

Stamati Morellas

COM S 311 – Homework 2

Section 6 – Tuesday 1:10 – John Wahlig

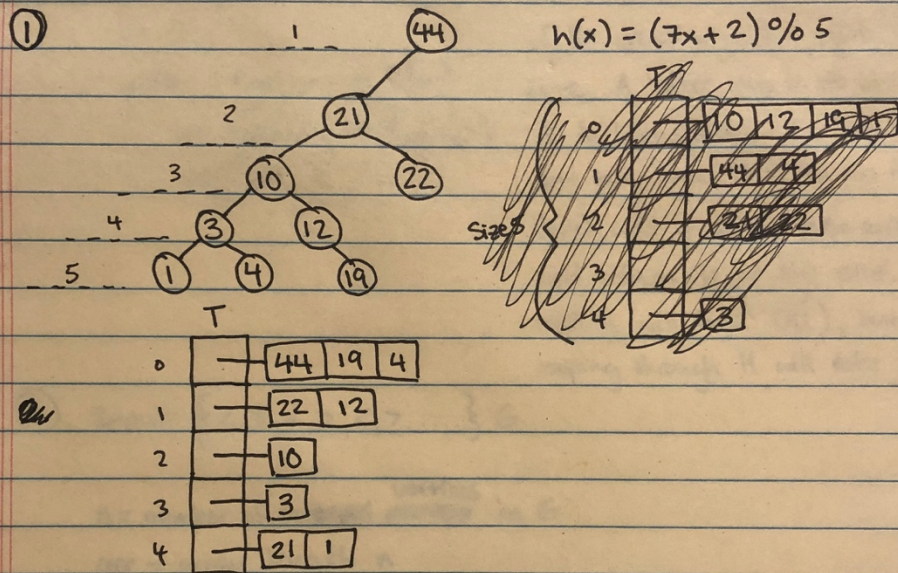
Due: 9/23/19

Stamati Morellas

Section 6 - Tuesday 1:10 - John Wahlig

9/22/19

①  $h(x) = (7x + 2) \% 5$



② a. function  $\text{succ}(H, x)$   $\text{succ}(x)$  - Smallest element in S that is greater than x

```

next = x + 1 // next smallest element
while (next < H.size)
    if H[next] is not null
        add - O(1)
        delete - O(1)
        search - O(1)
        return H[next]
    else
        next++
return null // if nothing is found
  
```

The runtime should be  $O(n)$  which is the amount of times gone through the while loop (iterations)



②b.

$n = \text{num elements in } H$   
 $\text{arr} = \text{new array length } n$   
 $x = -\infty$   
 $i = 0$

while  $i < n \rightarrow O(n)$   
 $x = \text{succ}(x) \rightarrow O(n)$   
 $\text{arr}[i++] = x$

Since  $\text{succ}(x)$  takes  $O(n)$  time and the while loop that encloses it also takes  $O(n)$  time, the runtime of the overall algorithm takes  $O(n^2)$  time. A better way to do this is to loop through the hash table and every time there is an integer (key), add it to a new array and return it at the end. adding will take  $O(1)$ , but looping through  $H$  will take  $O(n)$ .

③ Input:  $\{<, >, <, >, <, >, \dots\} G$

$n = \text{number of } \text{vertices} \text{ in } G$   
 $\text{arr} = \text{array length } n$   
 $i = 0$  //  
 $j = 1$

while  $(i < n) \{$   
 while  $(j < n) \{$   
 ~~$\text{if } \text{arr}[i] \text{ and } \text{arr}[j] \text{ are inverse elements}$~~   
 $\text{if } \text{arr}[i] \text{ and } \text{arr}[j] \text{ are inverse elements}$   
 $\text{break inner loop}$   
 $\}$   
 $\}$   
 $\}$

This algorithm will take  $O(n^2)$  time because of the two while loops.



④

a) decreaseKey(index, delta) {

~~currentKey = H(index).Key~~  
~~currentKey -= delta~~  
~~if currentKey < currentKey.Parent~~  
~~heapify up~~  
}

currentKey = H(index).Key

currentKey -= delta

if currentKey < currentKey.Parent

heapify up

}

~

The worst-case runtime of this algorithm is  $O(\log n)$  if the function runs the if-statement

b) Hash Tables are the only data structure that

have a search, insertion, and deletion time of  $O(1)$ , so implementing a hash table where the key value pairs include the key and its respective index should allow for the best results here.