**Sam Lazrak**

**CS 303 Algorithms and Data Structures**

**Homework Assignment 6**

**2/20/17**

1. Work the following Exercise from Chapter 10 of the text:
   1. (2 points) Exercise 10.1-4 on page 235.

* ENQUEUE(Q, x)

if Q.head == Q.tail +1

error “overflow”

else

Q[Q.tail] = x

if Q.tail == Q.length

Q.tail = 1

else Q.tail = Q.tail + 1

* DEQUEUE(Q)

if Q.tail == Q.head

error “underflow”

else

x = Q[Q.head]

if Q.head == Q.length

Q.head = 1

else Q.head = Q.head + 1

return x

* 1. (2 points) Exercise 10.2-2 page 240.
* PUSH(L, x)

Node top = y

x.prev = y

top = x

* POP(L)

Node temp = top

top = top.prev

return top

* 1. (2 points) Exercise 10.2-3 page 240.
* ENQUEUE(L, x)

Node last = y

x.prev = y

last = x

* DEQUEUE(L)

Node temp = first

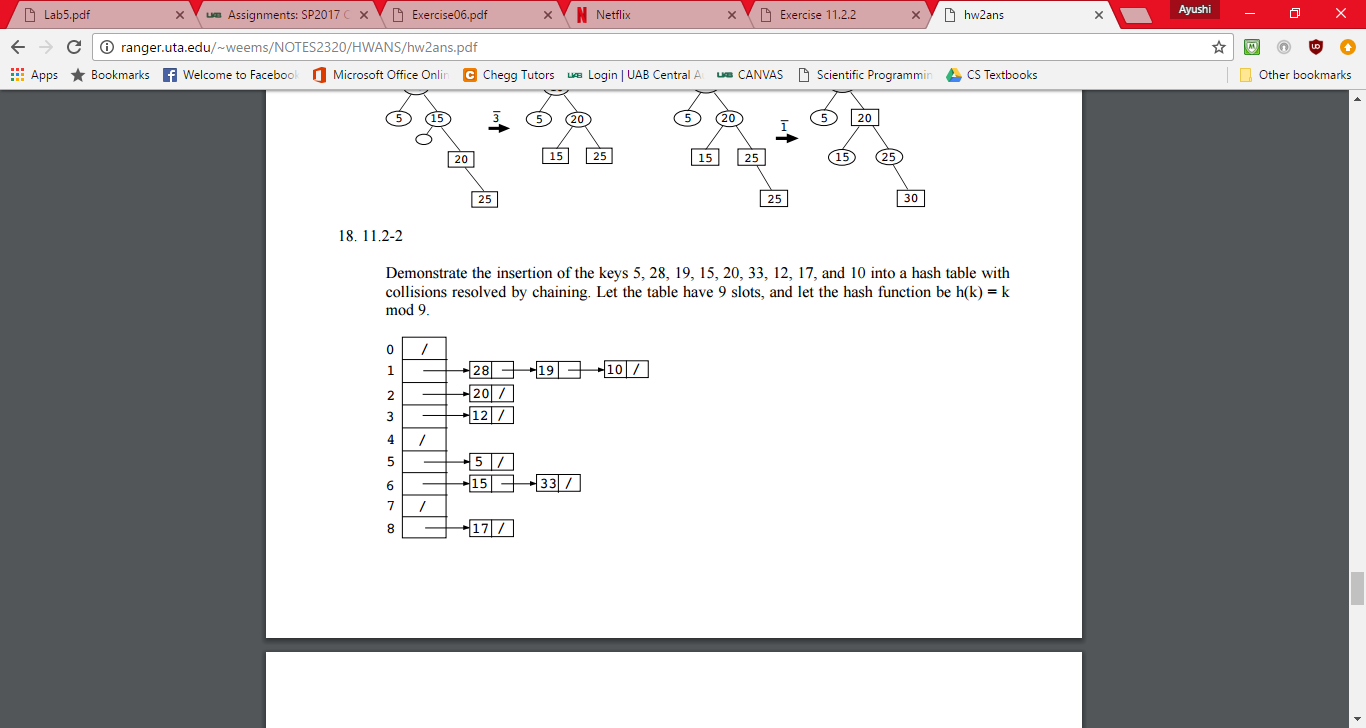
first = first.next

return temp

* 1. (5 points) Problem 10-1 page 249.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | unsorted, singly linked | sorted,  singly linked | unsorted, doubly linked | sorted, doubly linked |
| SEARCH(L, k) |  |  |  |  |
| INSERT(L, x) |  |  |  |  |
| DELETE(L, x) |  |  |  |  |
| SUCCESSOR(L, x) |  |  |  |  |
| PREDECESSOR(L, x) |  |  |  |  |
| MINIMUM(L) |  |  |  |  |
| MAXIMUM(L) |  |  |  |  |

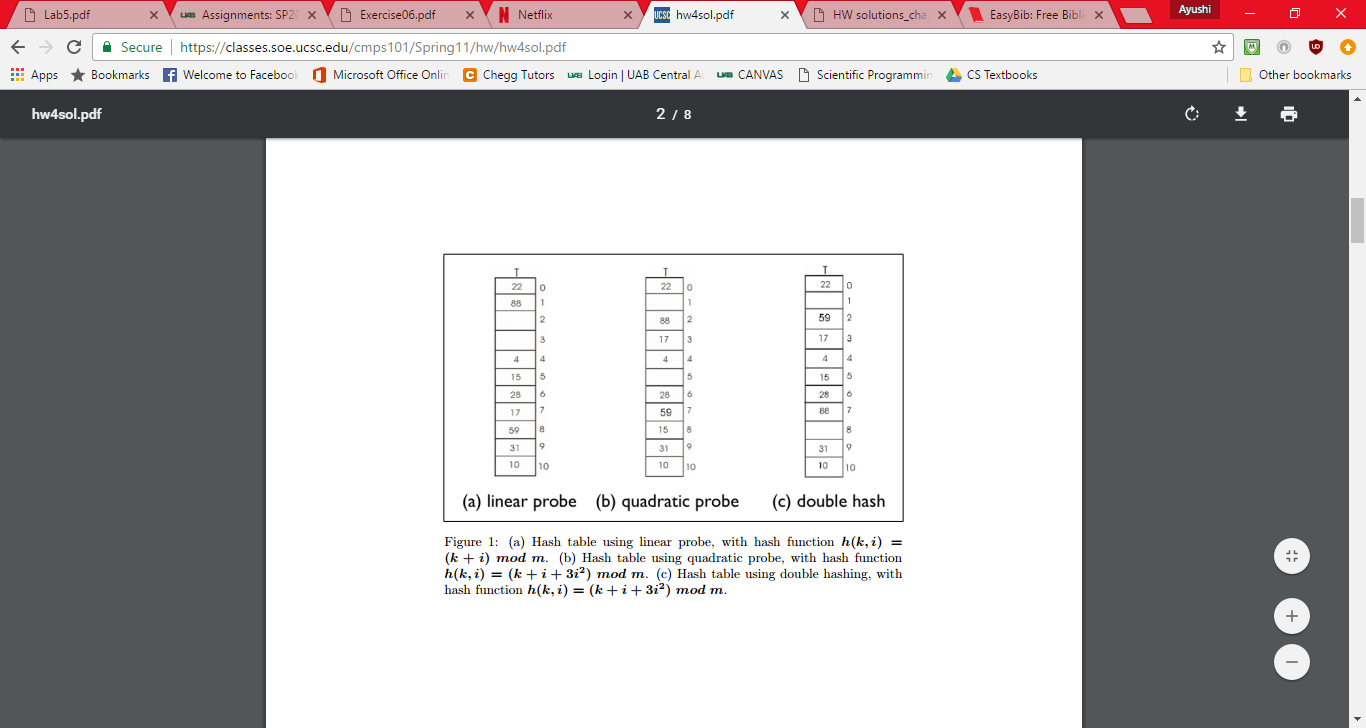
1. Work the following Exercises from Chapter 11 of the text:
   1. (4 points) Exercise 11.2-2, page 261.



(Kung)

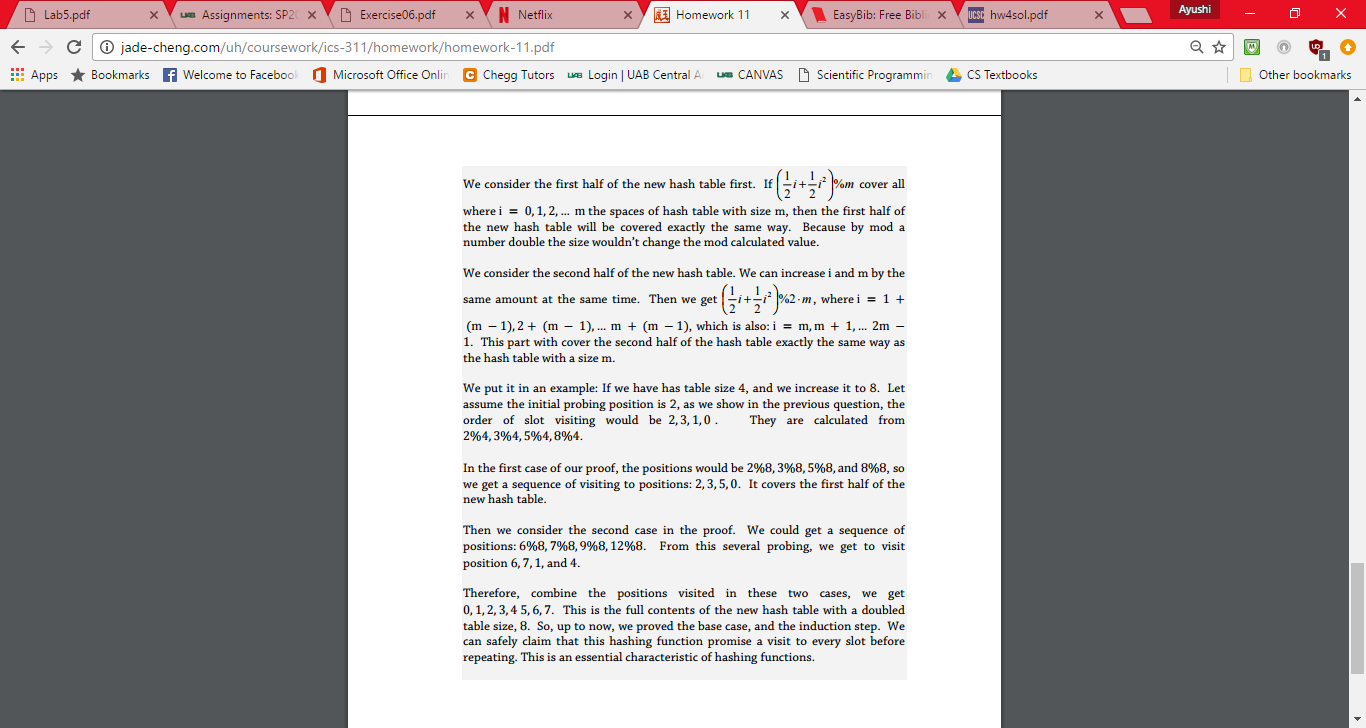
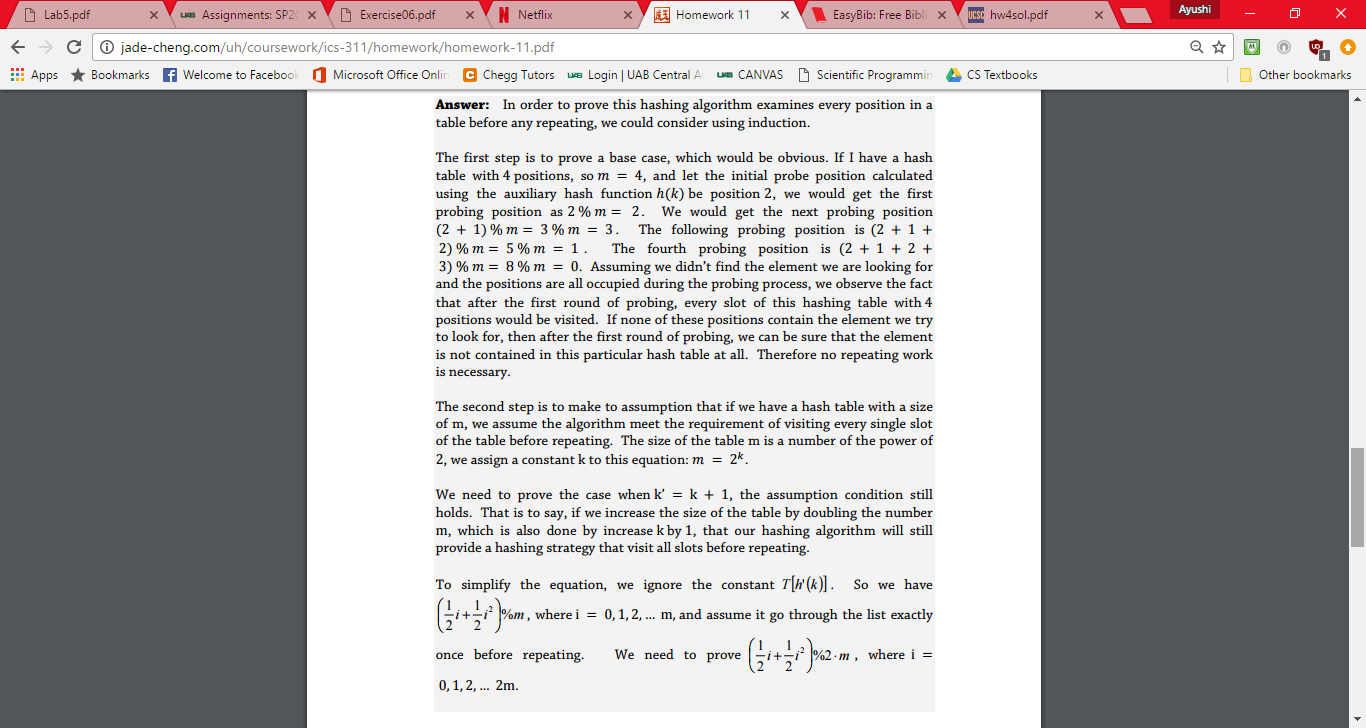
* 1. (2 points) Exercise 11.2-3, page 261.
* Successful searches: Θ(1 + α), which is identical to the original running time. The element we search for is equally likely to be any of the elements in the hash table, and the proof of the running time for successful searches is similar to what we did in the lecture.
* Unsuccessful searches: 1/2 of the original running time, but still Θ(1 + α), if we simply assume that the probability that one element's value falls between two consecutive elements in the hash slot is uniformly distributed. This is because the value of the element we search for is equally likely to fall between any consecutive elements in the hash slot, and once we find a larger value, we can stop searching. Thus, the running time for unsuccessful searches is a half of the original running time. Its proof is similar to what we did in the lecture.
* Insertions: Θ(1 + α), compared to the original running time of Θ(1). This is because we need to find the right location instead of the head to insert the element so that the list remains sorted. The operation of insertions is similar to the operation of unsuccessful searches in this case.
* Deletions: Θ(1 + α), same as successful searches.

("Homework 11 Solutions.")

* 1. (6 points) Exercise 11.4-1, page 277.

("Homework 4 Solutions.")

* 1. (6 points) Problem 11-3, page 283. Do part (b) only.



(Cheng, 2008)

Works Cited

Cheng, Jade Yu. *Homework 11* (n.d.): 3-4. 9 Oct. 2008. Web.

"Homework 4 Solutions." *Texts in Computer Science Theory of Computation* (n.d.): 329-31. Web.

"Homework 11 Solutions." *Texts in Computer Science Theory of Computation* (n.d.): 354-56. 2013. Web.

Kung, Sidney H. "Sums of Powers." *The Mathematics Teacher* 95.9 (2002): 695. Web.