Workshop 5

Exercise 1 Skip Lists

Problem 1. Recall from lectures that the height of a list at the insertion of an element into a skip-list is h where h is the number of flips of a coin that it takes until a head is achieved.

Given the information above, draw the development of a skip-list resulting from the insertion of the elements:

20, 11, 17, 3, 45, 76, 23, 14

assume that the corresponding sequence of coin flips is:

H T T H H T H T H T T T H T H H T T H H H T T T T H T H

where H stands for heads and T stands for tails. Show the skip-list after each insertion.

Exercise 2 Hash Tables

Problem 1. Insert the key sequence 7, 18, 2, 3, 14, 25, 1, 11, 12, 1332 with hashing by chaining in a hash table with size 11. Please show the final table by using the hash function h(k) = k mod 11

Problem 2. Please show the final table if we use linear probing instead.

Problem 3. Investigate by yourself what is "quadratic probing" and "double hashing". Both can be considered improved versions of linear probing. Please find out where they improve upon linear probing.

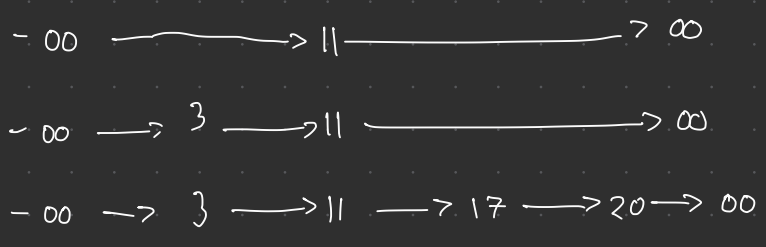
Exercise 1 - Skip Lists

insert (20) H = 1

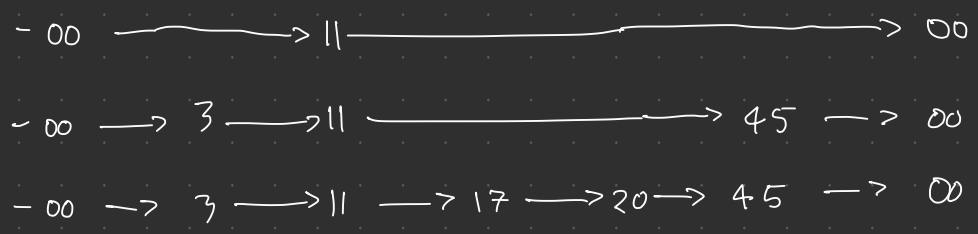
insert (11) H = 3

insert (17) H = 1

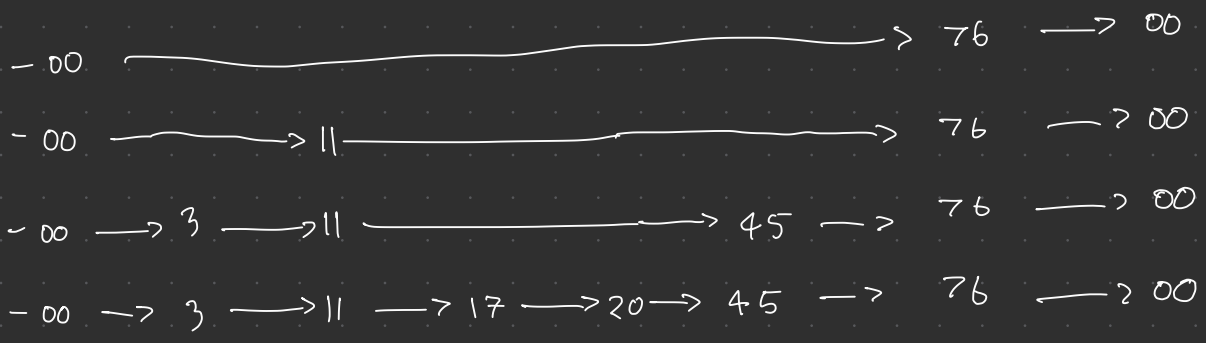
insert (3) H = 2



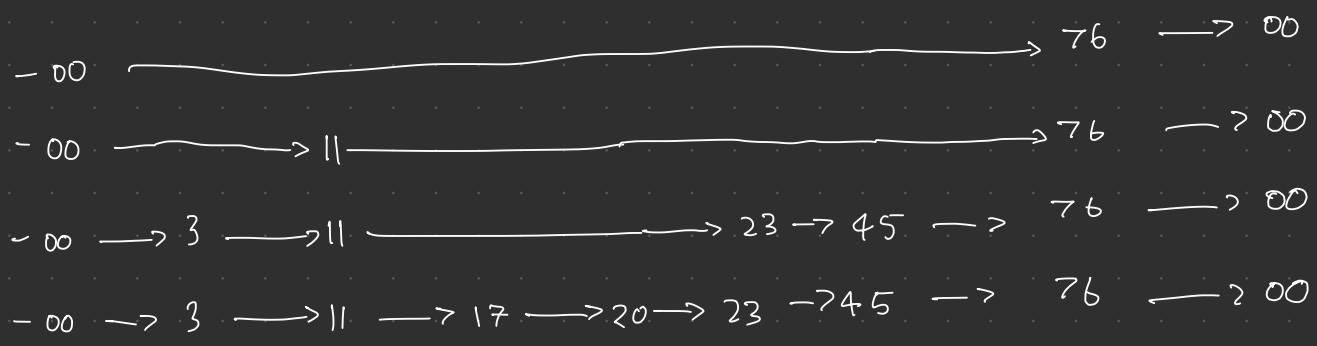
insert (45) H=2



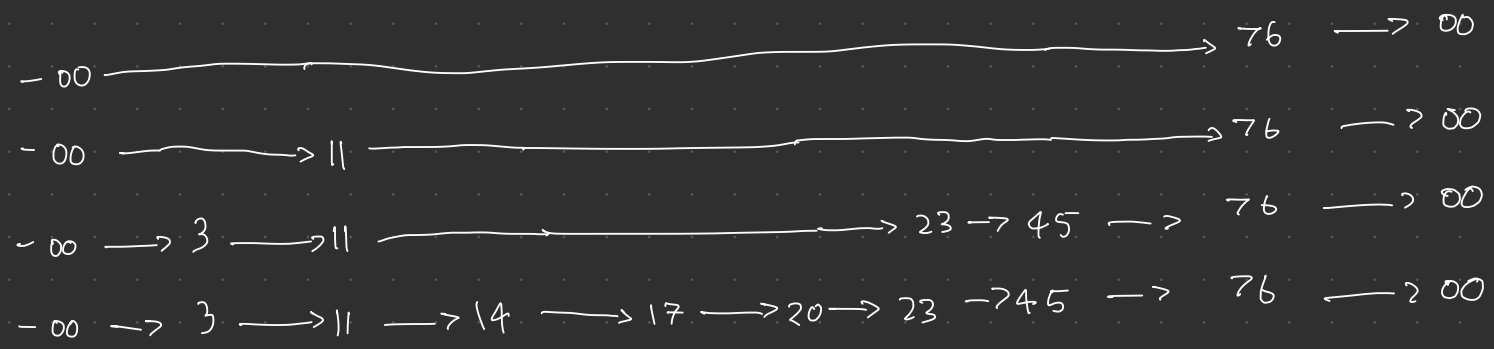
insert (76) H=4



insert (23) H=2



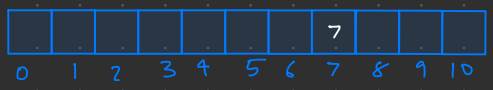
insert (14) H=1



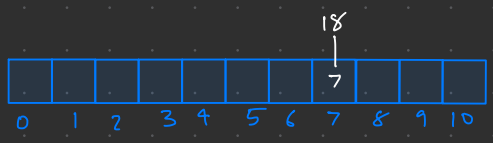
Exercise 2 - Hash Tables

1. $h(k) = k \bmod 11$, chaining

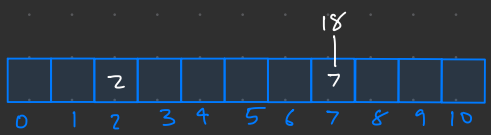
insert(7), key = 7



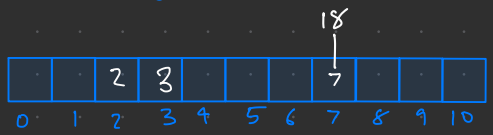
insert 18, key = 7



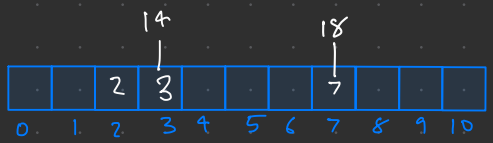
insert 2, key = 2



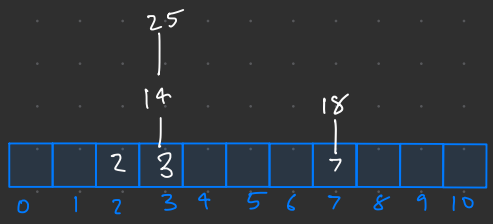
insert 3, key = 3



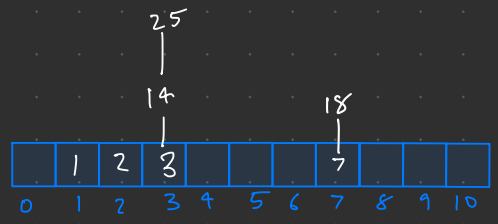
insert 14, key = 3



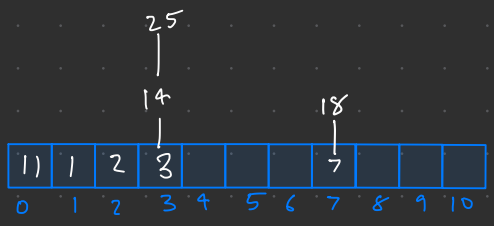
insert 25, key = 3



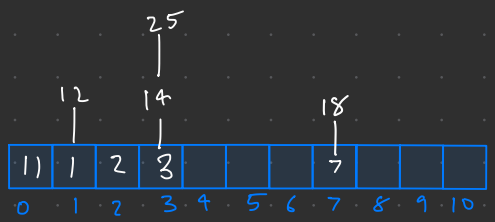
insert (1), key = 1



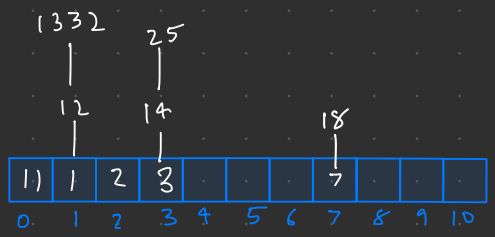
insert (11), key = 0



insert (12), key = 1

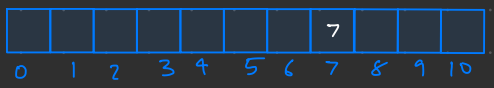


insert (1332), key = 1

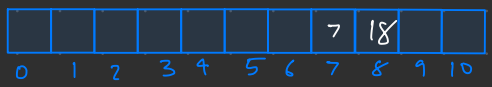


2. $h(k) = k \bmod 11$, linear probing

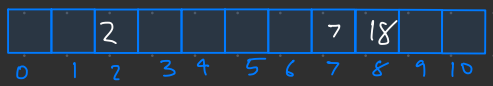
insert(7), key = 7



insert 18, key = 7

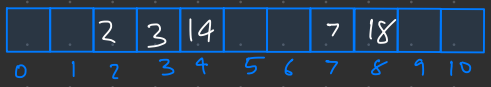


insert 2, key = 2

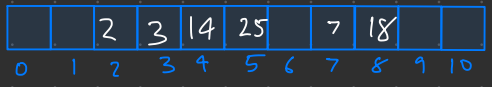


insert 3, key = 3

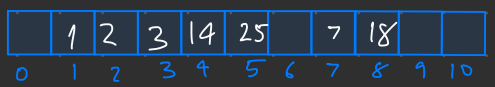
insert 14, key = 3



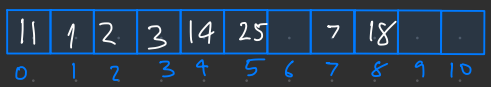
insert 25, key = 3



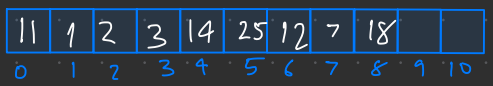
insert(1), key = 1



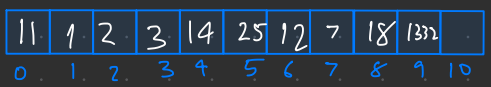
insert (11), key = 0



insert (12), key = 1



insert (1332), key = 1



3. quadratic probing - instead of inserting directly in the next available slot, we find the next slot using a quadratic addition depending on how many numbers have already been hashed

eg. hash function = h % 10

insert 5, key = 5 → no collisions

insert 25, key = 5 → 1 collision.

$index = 5 + 1^2$

insert 255, key = 5 → 2 collisions

$index = 5 + 2^2$

double hashing - using another hash function if collision occurs

eg insert 5, key = 5 (h % 10) → no collision

insert 25, key = 5 (h % 10) → collision

insert 25, key = 1, (h % 24) → no collision