# EL9343 Homework 06

**Spring 2022**

**Name:** Sagar Patel
**NETID:** sp5894

## Question 01

Demonstrate what happens when we insert the keys $26, 10, 20, 39, 2, 35, 19, 8, 22, 5$into a hash table with collisions resolved by chaining. Let the table have $9$ slots, and let the hash function be $h(k) = k \bmod 9$.

### Solution

The hash table is as shown —

| h(k) | key(s) |
|---|---|
| 0 | |
| 1 | 19 -> 10 |
| 2 | 2 -> 20 |
| 3 | 39 |
| 4 | 22 |
| 5 | 5 |
| 6 | |
| 7 | |
| 8 | 8 -> 35 -> 26 |

## Question 02

Exercise 11.2-1 in CLRS Textbook.

### Solution

Under the assumption of simple uniform hashing, we will be using the linearity of expectation to compute it – assuming that all the keys are ordered in the format $\{k_1, k_2, ..., k_n\}$. Let $X_i$ be the number of $1 > k_i$ such that $h(1) = h(k_i)$. Note that it is the same as $\sum_{j>i} P(h(k_j) = h(k_i)) \Rightarrow \sum_{j>i} \frac{1}{m} \Rightarrow \frac{(n-i)}{m}$. By the linearity of expectations, the number of collisions is the sum of number of collisions for each possible smallest element in the collision.

Therefore, the expected number of collisions will be – $\sum_{i=1}^{n} \frac{n-i}{m} = \frac{n^2 - \frac{n(n+1)}{2}}{m} = \frac{n^2-n}{2m}$

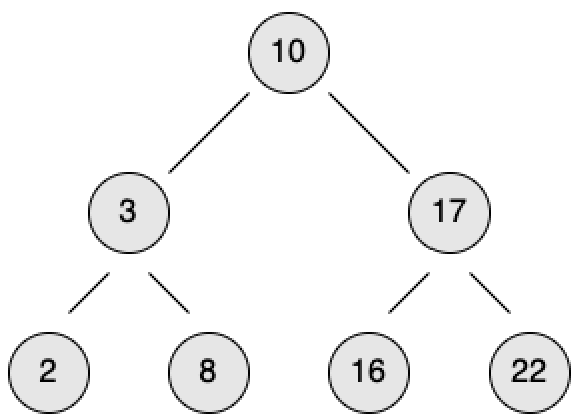
## Question 03

For the set of $\{2, 3, 8, 10, 16, 17, 22\}$ of keys, draw binary search trees of heights $2, 3, 4, 5,$ and $6.$

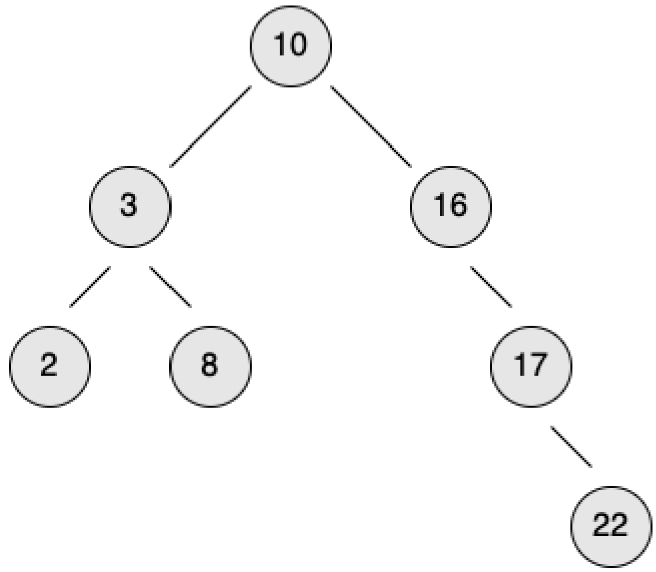
### Solution

NOTE: All the diagrams have been drawn using draw.io

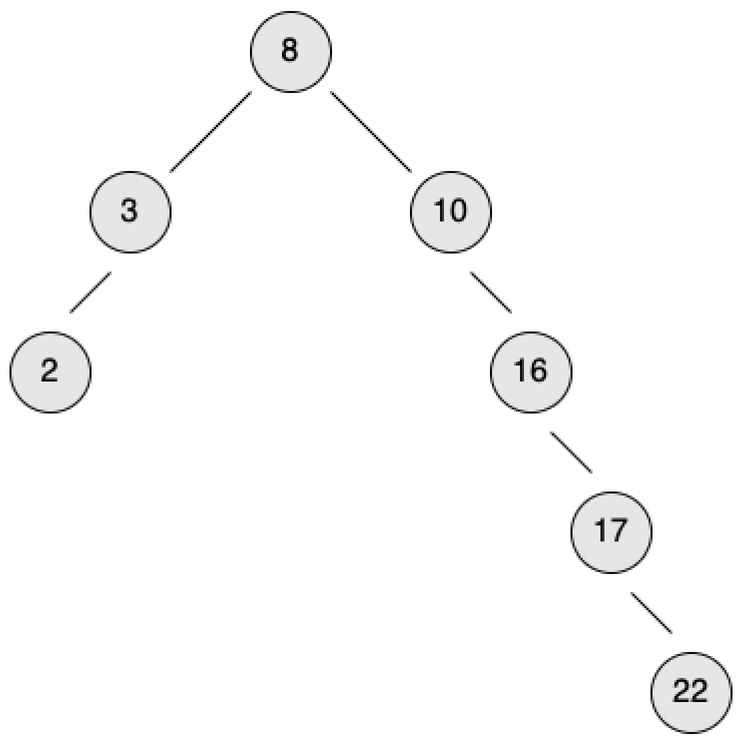
-- For height **h=2**



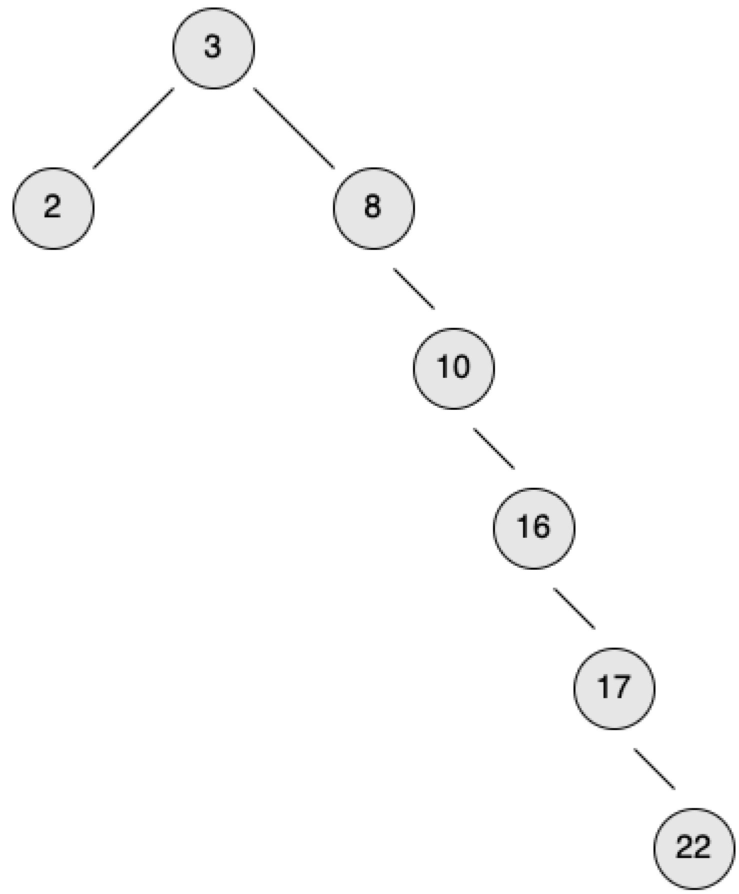

-- For height **h=3**



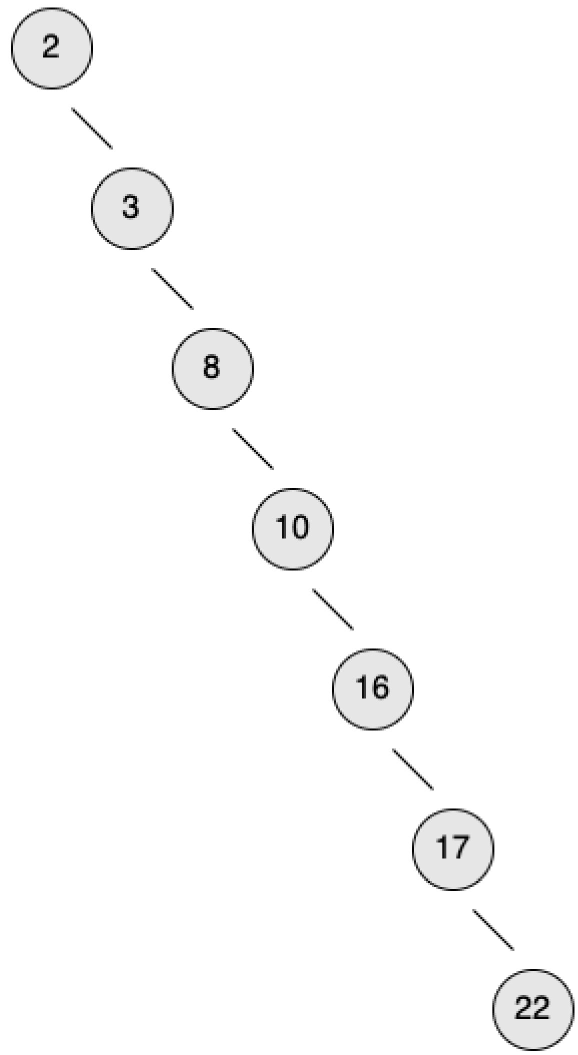

-- For height **h=4**




-- For height **h=5**

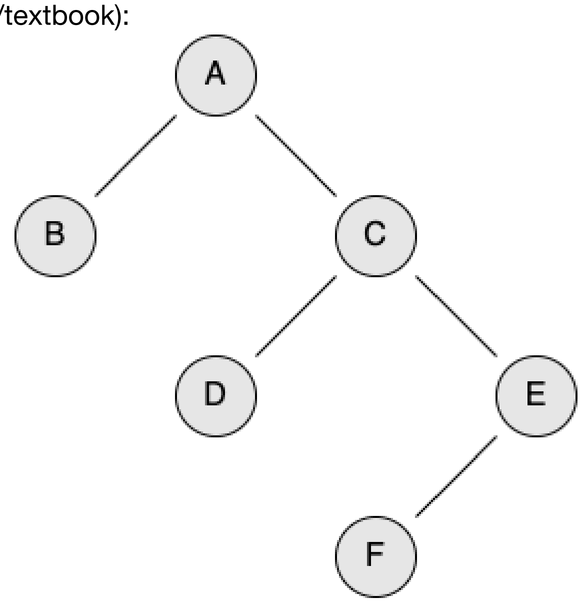



-- For height **h=6**




## Question 04

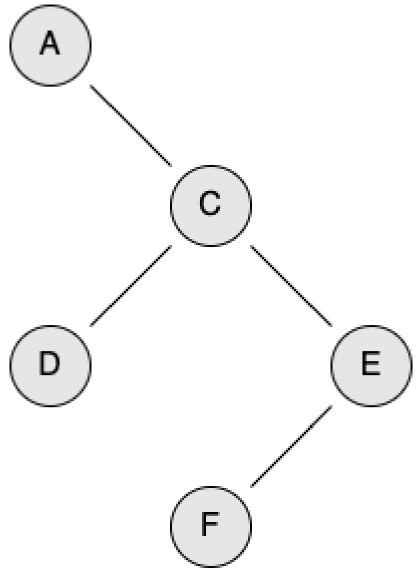
For the following binary search tree, show the result of following operations (Please follow the algorithm from the lecture/textbook):




**4.a**

Delete B.

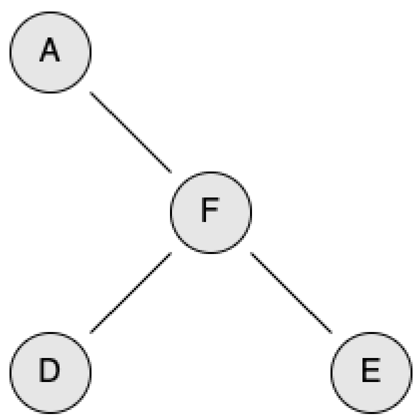
## Solution




#### 4.b

Delete C from the result of 4.a

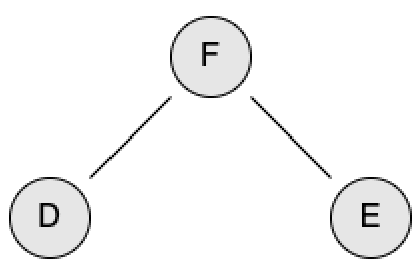
## Solution




#### 4.c

Delete A from the result of 4.b

## Solution



**4.d**

Delete A from the original tree.

## Solution