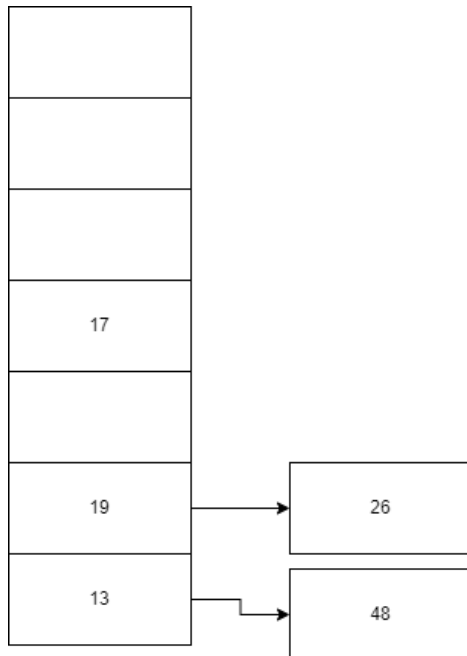


Question 1

a. Separate chaining



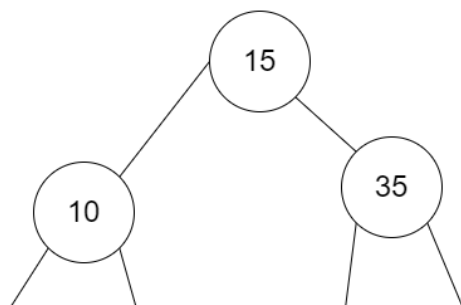
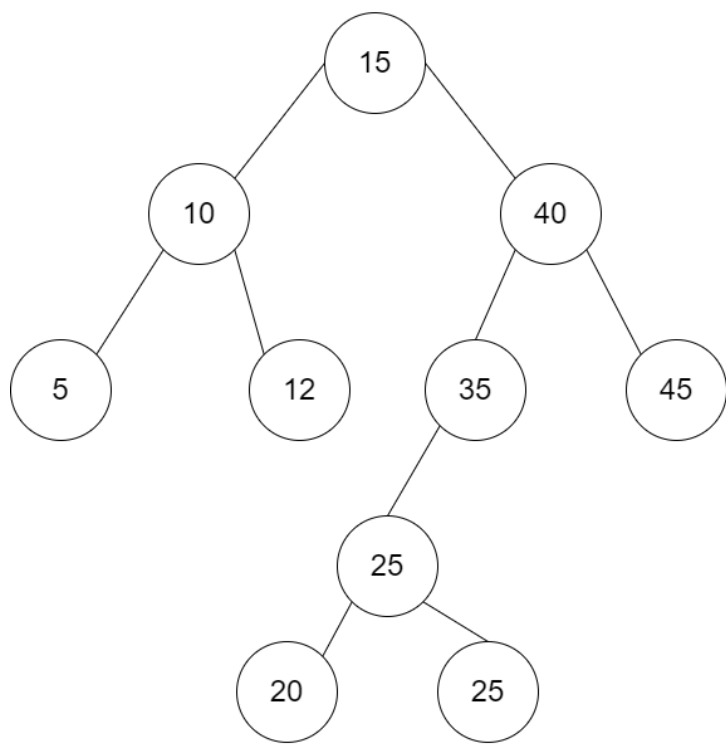
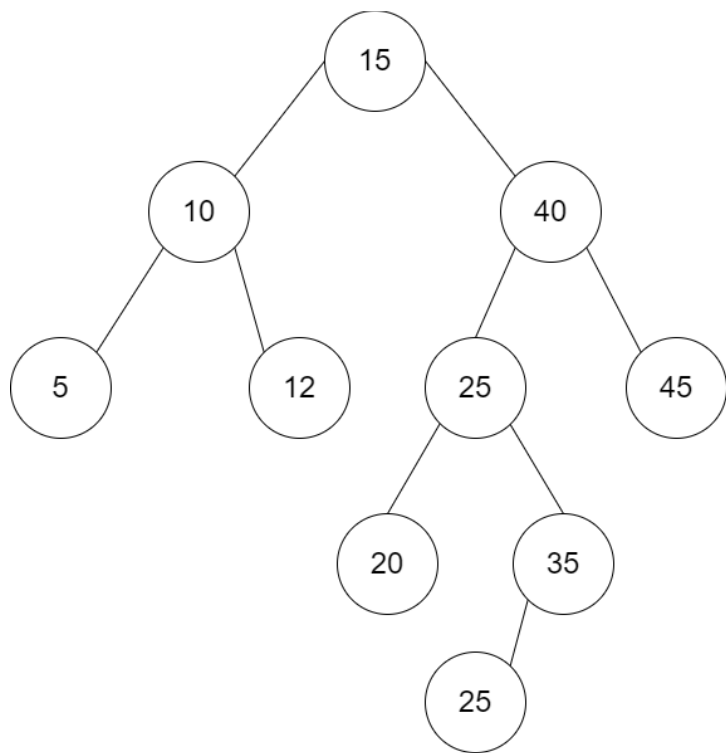
b. Linear probing



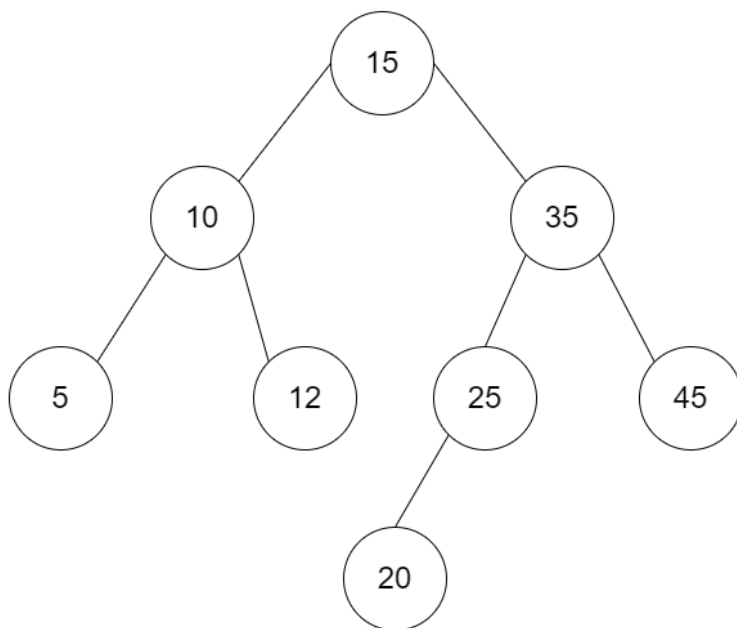
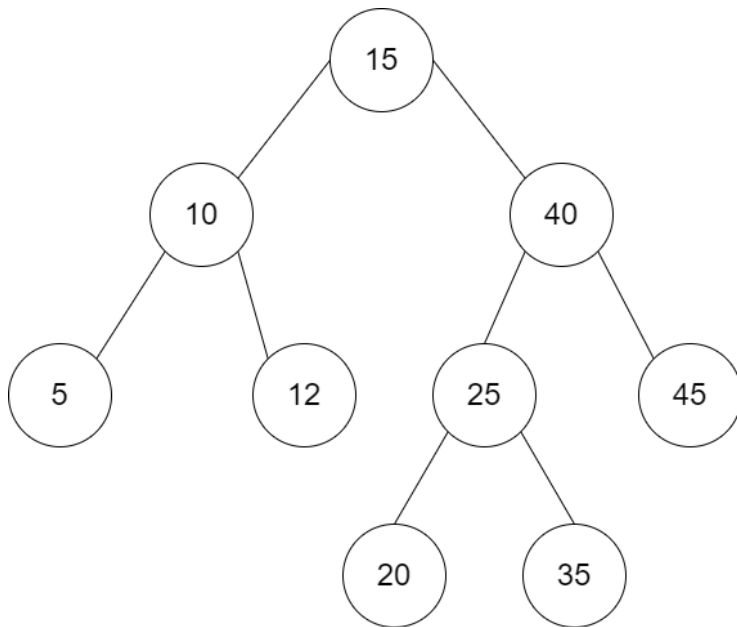
c. Double hashing

48
17
26
19
13

Question2
Insert key 25



b) remove 40



Question 3

```
def double_hashing(keys, hashtable_size, double_hash_value):  
    hashtable_list = [None] * hashtable_size  
    for i in range(len(keys)):  
        hashkey = keys[i] % hashtable_size  
        if hashtable_list[hashkey] is None:  
            hashtable_list[hashkey] = keys[i]  
        else:
```

```
new_hashkey = hashkey
while hashtable_list[new_hashkey] is not None:
    new_hashkey = random_value(new_hashkey)
hashtable_list[new_hashkey] = keys[i]
return hashtable_list
```

- a) This is not a more efficient approach compared to double hashing. Because a new key generated by a random generator may still collide with an existing key, double hashing has more control about the new key to avoid collide again.
- b) Double hashing is preferred.