

**2018 SS**

The following 2 questions will use this hashtable of size 8. The hash function used to insert the keys is  $h(key) = key \% 8$ . Insert the following 5 key values into the empty hashtable in order from left to right.

3, 6, 10, 24, 37

Hashtable:  $h(key) = key \% 8$

0	1	2	3	4	5	6	7

**Question 31**

[1 mark] What key value will be mapped to index 5 after all values have been inserted?

- (a) 10
- (b) 24
- (c) 6
- (d) 37
- (e) None of the above

**Question 32**

[1 mark] What key value will be mapped to index 0 after all values have been inserted?

- (a) 3
- (b) 24
- (c) 10
- (d) 37
- (e) None of the above

The following 2 questions will use this hashtable of size 8. Insert the following 5 key values into the empty hashtable in order from left to right. Solve any collisions by using **Linear Probing**.

3, 11, 24, 35, 37

Hashtable:  $h(key) = key \% 8$

0	1	2	3	4	5	6	7

**Question 33**

[1 mark] What key value will be mapped to index 5 after all values have been inserted?

- (a) 35
- (b) 11
- (c) 3
- (d) 37
- (e) None of the above

**Question 34**

[1 mark] What key value will be mapped to index 6 after all values have been inserted?

- (a) 11
- (b) 35
- (c) 37
- (d) 24
- (e) None of the above

Insert the following 5 key values into the empty hashtable in order from left to right. Solve any collisions by using **Quadratic Probing**.

3, 12, 19, 22, 28

Hashtable:  $h(key) = key \% 8$

0	1	2	3	4	5	6	7

### Question 35

[1.5 marks] What key value will be mapped to index 5 after all values have been inserted?

- (a) 19
- (b) 22
- (c) 12
- (d) 28
- (e) None of the above

Insert the following 5 key values into the empty hashtable in order from left to right. Solve any collisions by using **double hashing** where the second hash function is defined as:

$$h'(key) = 5 - (key \% 5)$$

3, 12, 19, 23, 25

Hashtable:  $h(key) = key \% 8$

0	1	2	3	4	5	6	7

### Question 36

[1.5 marks] What key value will be mapped to index 6 after all values have been inserted?

- (a) 19
- (b) 3
- (c) 23
- (d) 25
- (e) None of the above

### 2017 S2

#### Question 15

[2 marks] For this question **linear probing** is used to resolve collisions in a hash table. The hash table is of size 7 and is initially empty.

\_ ' \_ ' \_ ' \_ ' \_ ' \_

The following FIVE key values 34, 46, 11, 63, 19 are inserted into the hash table in the order shown from left to right. The hash function used to insert these keys is

$$h(key) = key \% 7$$

Which of the following shows the resulting table?

- (a) 63, \_, \_, \_, 11, 19, 34
- (b) 63, \_, \_, \_, 46, 11, 34
- (c) 63, 19, \_, \_, 46, 11, 34
- (d) 63, \_, \_, 11, 46, 19, 34
- (e) None of the above.

**Question 16**

[2 marks] For this question **plus 5 probing** is used to resolve collisions in a hash table. The hash table is of size 7 and is initially empty.

\_ ' \_ ' \_ ' \_ ' \_ ' \_

The following FIVE key values 59, 38, 17, 27, 43 are inserted into the hash table in the order shown from left to right. The hash function used to insert these keys is

$$h(\text{key}) = \text{key} \% 7$$

Which of the following shows the resulting table?

- (a) \_, 43, 17, 59, \_, 27, 38
- (b) \_, 43, \_, 59, \_, \_, 27
- (c) 38, 43, 59, 27, \_, 17, \_
- (d) \_, 38, 43, 59, 27, \_, 17
- (e) None of the above.

**Question 17**

[2 marks] For this question **quadratic probing** is used to resolve collisions in a hash table. The hash table is of size 7 and is initially empty.

\_ ' \_ ' \_ ' \_ ' \_ ' \_

The following FIVE key values 15, 22, 71, 64, 10 are inserted into the hash table in the order shown from left to right. The hash function used to insert these keys is

$$h(\text{key}) = \text{key} \% 7$$

Which of the following shows the resulting table?

- (a) \_, 64, \_, 10, \_, \_, \_
- (b) \_, 15, 22, 64, 10, 71, \_
- (c) 15, 22, 64, 10, 71, \_, \_
- (d) \_, 15, 22, 10, \_, 71, \_
- (e) None of the above.

**Question 18**

[2 marks] For this question **double hashing** is used to resolve collisions in a hash table. The hash table is of size 7 and is initially empty.

\_ ' \_ ' \_ ' \_ ' \_ ' \_

The following FIVE key values 51, 79, 37, 16, 47 are inserted into the hash table in the order shown from left to right. The hash function used to insert these keys is

$$h_1(\text{key}) = \text{key} \% 7$$

The second hash function used to determine the skip number is

$$h_2(\text{key}) = 6 - (\text{key} \% 5 + 1)$$

Which of the following shows the resulting table?

- (a) \_, \_, 51, 79, 37, 16, 47
- (b) \_, 79, 51, 37, 16, 47, \_
- (c) \_, \_, 51, \_, \_, 47, \_
- (d) \_, 47, 51, 79, \_, 37, 16
- (e) None of the above.

**2017 S1**

**Question 35**

[2 marks] For this question linear probing is used to resolve collisions in a hash table. The hash table is of size 7 and is initially empty:

[None, None, None, None, None, None, None]

The following FIVE key values 25, 19, 32, 6, 12 are inserted into the hash table in the order shown from left to right. The hash function used to insert these keys is

$$h(\text{key}) = \text{key} \% 7$$

Which of the following shows the resulting table?

- (a) [6, 32, None, None, 25, 12, 19]
- (b) [12, None, None, 25, 19, 32, 6]
- (c) [6, 12, None, None, 25, 19, 32]
- (d) [12, None, None, 25, 32, 19, 6]
- (e) None of the above.

**Question 36**

[2 marks] For this question quadratic probing is used to resolve collisions in a hash table. The hash table is of size 7 and is initially empty:

[None, None, None, None, None, None, None]

The following FIVE key values 25, 19, 32, 6, 12 are inserted into the hash table in the order shown from left to right. The hash function used to insert these keys is

$$h(\text{key}) = \text{key} \% 7$$

Which of the following shows the resulting table?

- (a) [None, 32, 12, None, 25, 19, 6]
- (b) [12, None, None, 25, 19, 32, 6]
- (c) [6, 12, None, None, 25, 19, 32]
- (d) [32, None, None, 25, 19, 12, 6]
- (e) None of the above.

**Question 37**

[2 marks] For this question double hashing is used to resolve collisions in a hash table. The hash table is of size 7 and is initially empty:

[None, None, None, None, None, None, None]

The following FIVE key values 25, 19, 32, 6, 12 are inserted into the hash table in the order shown from left to right. The hash function used to insert these keys is

$$h_1(\text{key}) = \text{key} \% 7$$

The second hash function used to determine the step size for probing is

$$h_2(\text{key}) = 5 - \text{key} \% 5$$

Which of the following shows the resulting table?

- (a) [6, None, None, 25, 19, 12, 32]
- (b) [None, 12, None, 25, 19, 32, 6]
- (c) [32, 12, None, None, 25, 19, 6]
- (d) [6, 12, None, None, 25, 19, 32]
- (e) None of the above.

**Question 38**

[2 marks] Given a simple Hash table class defined as below:

```
class BasicHashTable:
    def __init__(self, size=7):
        self.size = size
        self.slots = [None] * self.size
    def hash_function(self, key):
        return key % self.size
    def rehash(self, old_hash):
        return (old_hash + 1) % self.size
    def put(self, key):
        hash_value = self.hash_function(key)
        if self.slots[hash_value] == None:
            self.slots[hash_value] = key
        else:
            next_slot = self.rehash(hash_value)
            #DELETED_LINE
            next_slot = self.rehash(next_slot)
            if next_slot == hash_value:
                return
            if self.slots[next_slot] == None:
                self.slots[next_slot] = key
```

what code do we need to insert into **DELETED\_LINE** to make the put method work correctly?

- (a) while self.slots[next\_slot] == None and self.slots[next\_slot] != key:
- (b) while self.slots[next\_slot] != None and self.slots[next\_slot] != key:
- (c) while self.slots[next\_slot] == None and self.slots[next\_slot] == key:
- (d) while self.slots[next\_slot] != None and self.slots[next\_slot] == key:
- (e) None of the above

### Question 39

[2 marks] Which of the following four statements are true?

- I Hash tables are efficient both for simple operations (find, insert, delete) as well as more complex queries such as finding the k largest values or the values most similar to a given input.
- II The collision resolution technique “linear probing” often suffers from clustering
- III Clustering can be reduced by using double hashing
- IV Hashing with separate chaining always has  $O(1)$  performance

- (a) I and III
- (b) II, III and IV
- (c) I and II
- (d) II and III
- (e) All statements I-IV are true

### 2017 SS

#### Question 34

[8 marks]

The following **eight** key values:

12, 2, 5, 1, 20, 9, 10, 13

are to be inserted, in the order shown from left to right, into an empty hashtable of size 11. The hash function used to insert these keys is:

$$h(key) = key \% 11$$

On the diagram of the hashtable below, show the locations of the eight keys after they have been inserted. **Resolve any collisions using linear probing.**

Hashtable:  $h(key) = key \% 11$

0	1	2	3	4	5	6	7	8	9	10

Now, insert the same **eight** key values as before into a new, empty, hashtable of size 11. The same hash function as before is to be used. On the diagram of the hashtable below, show the locations of the eight keys after they have been inserted. **Resolve any collisions using quadratic probing.**

Hashtable:  $h(key) = key \% 11$

0	1	2	3	4	5	6	7	8	9	10

Finally, insert the same **eight** key values into a new, empty, hashtable of size 11. The same hash function as before is to be used. On the diagram of the hashtable below, show the locations of the eight keys after they have been inserted. **Resolve any collisions using double hashing with the secondary hash function:**

$$h'(key) = 5 - (key \% 5)$$

Hashtable:  $h(key) = key \% 11$ ;  $h'(key) = 5 - (key \% 5)$

## 2016S2

### Question 21

[2 marks] For this question **linear probing** is used to resolve collisions in a hash table. The hash table is of size 7 and is initially empty.

\_ \_ \_ \_ \_

The following FIVE key values 21, 13, 20, 27, 2 are inserted into the hash table in the order shown from left to right. The hash function used to insert these keys is

$$h(key) = key \% 7$$

Which of the following shows the resulting table?

- (a) 21, \_, 2, \_, \_, \_, 27
- (b) 21, \_, 2, \_, 27, 20, 13
- (c) 21, 13, 20, 27, 2, \_, \_
- (d) 21, 20, 27, 2, \_, \_, 13
- (e) None of the above.

### Question 22

[2 marks] For this question **quadratic probing** is used to resolve collisions in a hash table. The hash table is of size 7 and is initially empty.

\_ \_ \_ \_ \_

The following FIVE key values 15, 6, 9, 16, 23 are inserted into the hash table in the order shown from left to right. The hash function used to insert these keys is

$$h(key) = (2 * key + 3) \% 7$$

Which of the following shows the resulting table?

- (a) 9, 6, 23, \_, 16, 15, \_
- (b) 9, 6, 16, 23, \_, 15, \_
- (c) 15, 6, 9, 16, 23, \_, \_
- (d) 9, 6, \_, \_, \_, 15, \_
- (e) None of the above.

**Question 23**

[2 marks] For this question **double hashing** is used to resolve collisions in a hash table. The hash table is of size 7 and is initially empty.

— ' — ' — ' — ' — ' — ' —

The following FIVE key values 10, 17, 24, 3, 33 are inserted into the hash table in the order shown from left to right. The hash function used to insert these keys is

$$h_1(\text{key}) = \text{key} \% 7$$

The second hash function used to determine the skip number is

$$h_2(\text{key}) = 5 - \text{key} \% 5$$

Which of the following shows the resulting table?

(a) —, —, —, 10, —, 33, —

(b) 33, —, —, 10, 17, 24, 3

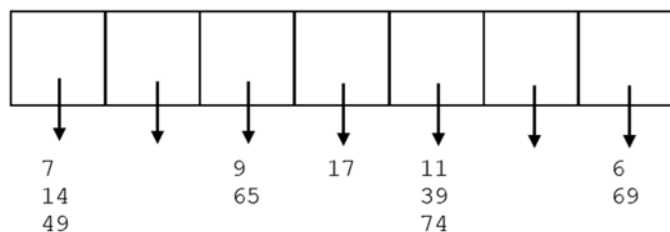
(c) —, 24, 3, 10, —, 33, 17

(d) 33, —, —, 10, 24, 3, 17

(e) None of the above.

**Question 24**

[2 marks] What is the load factor  $\lambda$  of the hash table shown below?



(a) 5/7

(b) 7/11

(c) 11/7

(d) 7/5

(e) Not possible to calculate given the provided information.

**Question 25**

[2 marks] Which of the following statements are **true**?

I A perfect hash function would uniformly map  $m$  distinct items into a hash table of size  $m$  with no collisions.

II A collision occurs when a hash function maps an item to an already occupied slot.

III The larger the load factor the greater the chance of collisions.

IV If a perfect hash function is used to insert  $m$  distinct items into a hash table of size  $m$ , searching for an item would always take  $O(1)$  time.

(a) I, II and IV

(b) I and II

(c) I, II and III

(d) II, III and IV

(e) All of the above.