What happens when you add entry to HashMap?

1) HashMap is created

a) Default Size : 16

b) Default Load Factor: 0.75

c) Entry Data type: <String, Integer>

2) Add an entry to the map

Syntax: map.put("Babu", 43);

a) "Babu" will be converted into hashcode --> 2062578

b) Find the index for the given hashcode --> 2

c) Add this entry to the index number 2:

1) Hash : 2062578

2) Key : Babu

3) Value : 43

4) Next Item : null

d) Size --> 1

3) Add another entry to the Map (Different Index)

Syntax: map.put("Hari", 40);

a) "Hari" will be converted into hashcode --> 2241808

b) Find the index for the given hashcode --> 0

c) Add this entry to the index number 0:

1) Hash : 2241808

2) Key : Hari

3) Value : 40

4) Next Item : null

d) Size --> 2

4) Add another entry to the Map (Collision)

Syntax: map.put("Haja", 28);

a) "Hari" will be converted into hashcode --> 2241552

b) Find the index for the given hashcode --> 0

c) Add this entry to the index number 0:

1) Hash : 2241552

2) Key : Haja

3) Value : 28

4) Next Item : null

d) Checks the existing hash of the entry in the index

1) Entry (Node) does exist

2) Check the Hash --> Hash code is different

3) Check the Next --> Next is null

e) Size --> 4

5) What is O-Notation for Put?

Best case : O(1)

Worst case: O(n) --> This is true until 1.7 (When LinkedList used)

Worst case: O(logn) --> This is true after 1.8 (When Tree used)

What happens when you search for entry to HashMap?

Pre-condition) You have a hashmap with filled entries

Step 1) Search for the entry

Syntax: map.get("Hari");

a) "Hari" will be converted into hashcode --> 2241808

b) Find the index for the given hashcode --> 0

c) Go to the index, search for Hari (key) with hash code !!

d) If both matches, retrieve the value and return

Step 2) Search for the entry that has collision

Syntax: map.get("Anil");

a) "Hari" will be converted into hashcode --> 2045488

b) Find the index for the given hashcode --> 0

c) Go to the index, search for Anil (key) with hash code !!

d) If it does not match, move the next node and repeat Step: c

d) If both matches, retrieve the value and return !

Step 3) What is the O Notation?

Best case : O(1)

Worst case: O(n) // 1.7 and O(logn) from 1.8

LinkedHashMap: O(n)

TreeMap: O(nlogn)