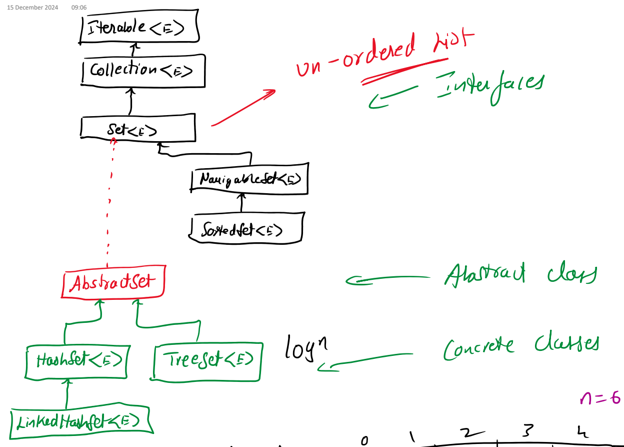


Collection

- add
- addAll
- contains
- remove
- removeAll
- retainAll
- size
- isEmpty
- iterator( )

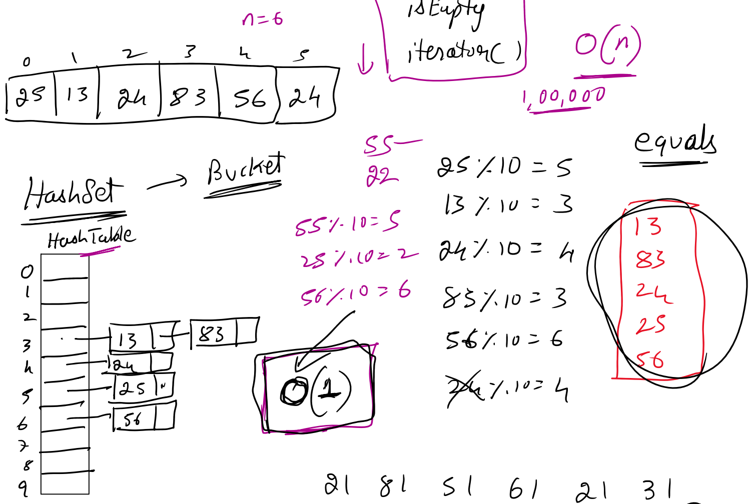
O(n)

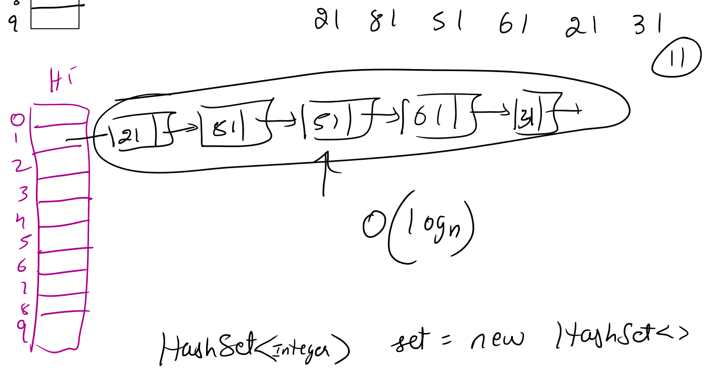
1,00,000

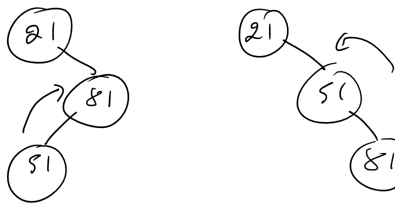
Ordered
ArrayList ✓
LinkedList ✓

Hash Table size
10

Hash Function
data % HTSize







HashSet<Integer> set = new HashSet<>

31

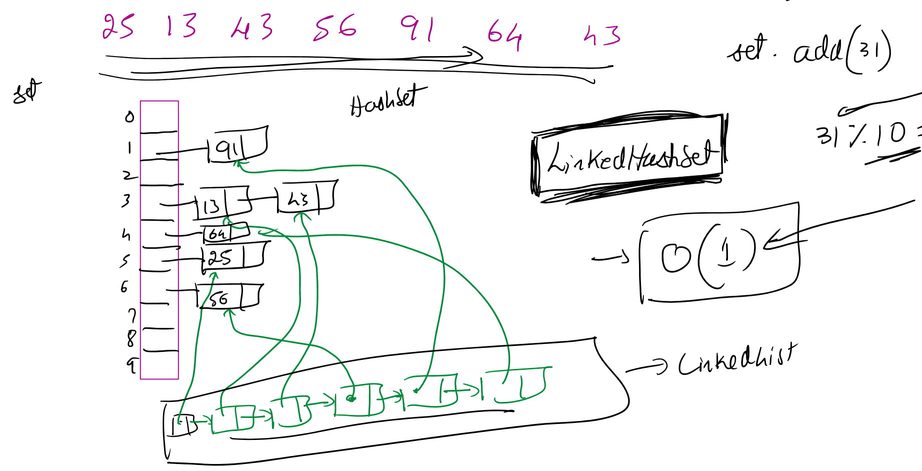
set.add(31)

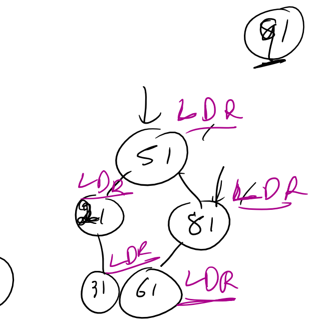
class Integer extends Object
{

public int hashCode( )
return 31

31 % 10 = 1

3
3





logn

Treeset

Inorder → 21
31
51
61
81

$\boxed{O(\log n)}$

02

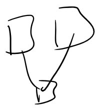
15 December 2024 09:07

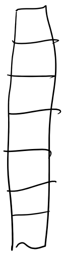
```
String longword = null;
for (String word : tree)
{
    if (longword == null)
        longword = word;
    else if (word.length() > longword.length())
        longword = word;
}
```

longword: Pineapple

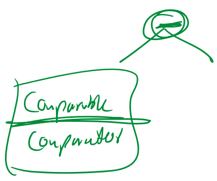
19 32 56 74 97

it



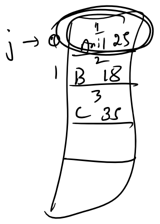


```
TreeSet<Employee> set;
```



```
Double.compare(this.cost, h2.cost);

public int compareTo(Hall h2)
{
    if (this.cost > h2.cost)
        return 1;
    elseif (this.cost < h2.cost)
        return -1;
    else
        return 0;
}
```



```
public void sort(ArrayList<E> list, Comparator<E> c)
{
    for (int i = 0; i < list.size() - 1 ; i++)
    {
        for (int j = 0; j < list.size() - 1; j++)
            if (c.compare(list.get(j), list.get(j+
        {
            ~~if (list.get(j).compareTo( list.ge~~
            {
                ≡
            }
        }
    }
}
```

$1)) > 0)$

$t(j+1)) > 0)$

03

15 December 2024 09:07

st="ABCABCDAB"     ABCD

0

```
String st= scan.nextLine()
LinkedHashSet<Character> set = new LinkedHashSet<Character>();
for(int i=0; i<st.length(); i++)
{   set.add(st.charAt(i))
}
for(Char ch: set)
    S.O.print(ch);
```

```
    0    1    2    3    4    5    6
arr 25 | 13 | 25 | 8 | 19 | 13 | 43
```

```
25  13  8  19  43
```

```
LinkedHashSet<Integer> set = new LinkedHashSet<>();
for(int i=0; i<arr.length; i++)
{
    set.add(arr[i]);
}
for(int data : set)
{
    System.out.print(data + " ");
}
```

```
class Recruitment implements Comparable<Recruitment>
{
    private String name;
    private String qualification;
    private String gender;
    private int exp;

    =
    public int compareTo(Recruitment r2)
    {
        if( r2.exp > this.exp)
            return 1;
        else if (r2.exp < this.exp)
            return -1;
        else
        {
            return r2.name.compareTo(this.name);
        }
    }
}
```

```
Class ExpNameComparator implements Comparator<Re
{
    public int compare(Recuritment r1, Recuri
    {
        if(r2.getExp() > r1.getExp())
            return 1;
        else if(r2.getExp() < r1.getExp
            return -1;

        else
        {
            return r2.getName().compare
        }
    }
}
```

Collections.sort(list);

ArrayList
LinkedList
Vector

Collections.sort(list, new ExpNameComparator());

~~HashSet~~
HashSet
LinkedHashSet
TreeSet

Comparable

Tre

Set

```
ArrayList<Integer> list= new ArrayList<>();
list.addAll(set);
```

```
ecruitment>

tment r2)

))

To(r1.getName());

eeSet<Employee> set = new TreeSet<>();

Class Employee implements Comparable<Employee>
{
```

↳ ArrayList<Integer> list = new ArrayList<>();

→ list.addAll(set);

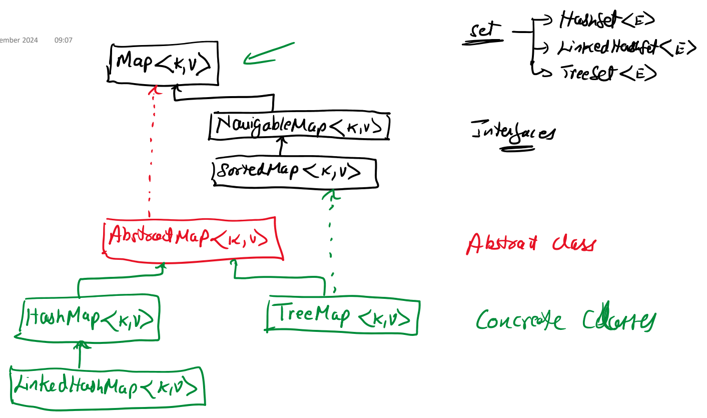
→ Collections.sort(list);

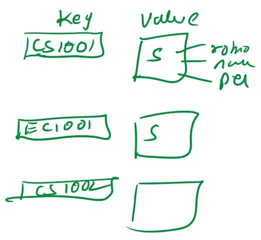
9 [illegible]

2

04

15 December 2024 09:07

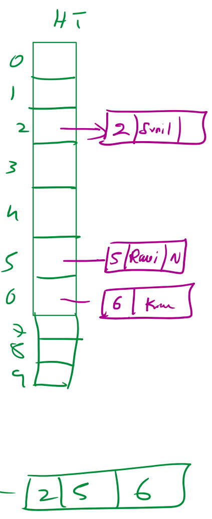


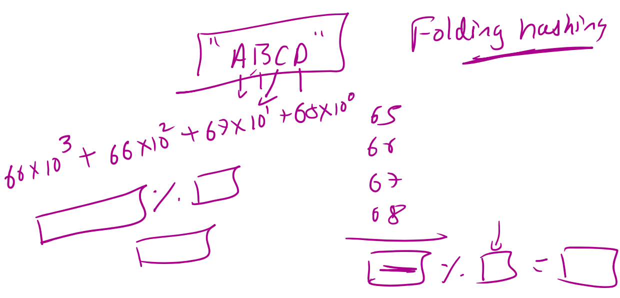


```
              Key        Value
HashMap<String, Student>  map = new HashMap<>();
map.put("CS1001", new Student(1, "Anil", 55));
map.put("EC1001", new Student(2, "Ravi", 66));
```

1) V put(K key, V value)
2) V get(K key)
3) set<K> keySet()
4) Collection<V> values()
5) boolean containsKey(K key)
6) boolean containsValue(V value)
7) int size()
8) boolean isEmpty()
9) V remove(K key)

```
                  Integer    String
null   <- map.put(2, "Anil")
Anil   <- map.put(2, "Sunil");
                2 % 10 = 2
Sunil  <- map.get(2);
map.put(5, "Ravi");
map.put(6, "Kiran")
```





```
"Ravi"  <- map.remove(5);
null    <- map.remove(8);
```

```
Set<Integer> set = map.keySet();
for (Integer key : set)
    S.O.P(map.get(key))
```

---

```
Collection<String> c = map.values();
for (String data : c)
    S.O.P(data);
```

# 05

15 December 2024 09:07