



# DSA – Data Structures

## HashTable



# Course Planning

Algorithms	Data Structures	Algorithmic Approaches	Interview Practices
1.Introduction	1.Asymptotic Analysis	1.Search Algorithms	1.In-place Reversal
2.Number 1	2.Dynamic Array	2.Sort Algorithms	2.Two Heaps
3.Number 2	3.LinkedList	3.Dac Algorithms	3.Subsets
4.String 1	4.Stack	4.Recursion	4.Modified BS
5.String 2	5.Queue	5.Sliding Window	5.Bitwise XOR
6.Array 1	6.HashTable	6.Two Pointers	6.Top 'K' Elements
7.Array 2	7.Tree	7.Fast & Slow	7.K-way Merge
8.Matrix	8.Trie	8.Cyclic Sort	8.Knapsack Problem
9.DP 1	9Directed Graph	9.Breadth First Search	9.Topological Sort
10.DP 2	10.Undirected Graph	10.Depth First Search	10.Mock Interview



# Implementation

Java – HashMap

JavaScript – Object

Python – Dictionary

C# - Dictionary

Swift - Dictionary

# java.util.HashMap

```
public class Main {  
    public static void main(String[] args) {  
        Map<Integer, String> map = new HashMap<>();  
        map.put(1, "Data");  
        map.put(2, "Structures");  
        map.put(3, "And");  
        map.put(4, "Algorithm");  
  
        System.out.println(map);  
    }  
}
```



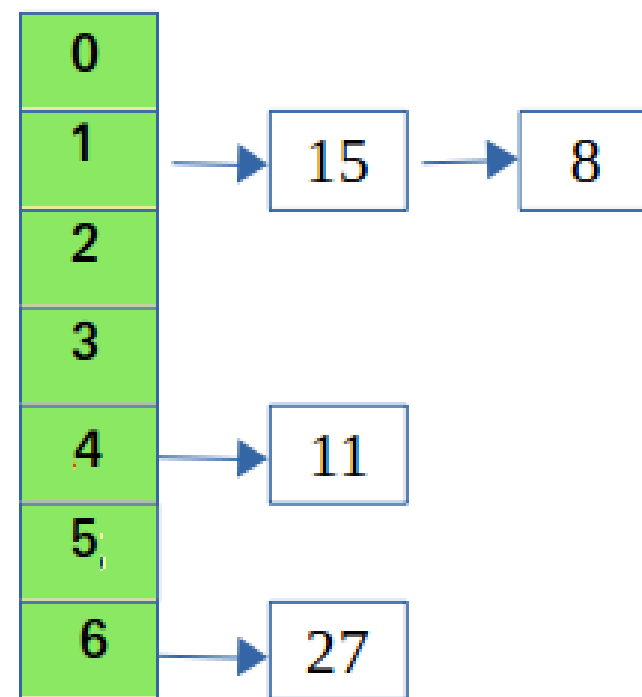
# Hash Function

```
public class Main {  
    public static void main(String[] args) {  
        Map<Integer, String> map = new HashMap<>();  
        map.put(9876543, "Data");  
  
        // array[1] = "Data"  
        // array => 100  
        // 0~99  
  
        System.out.println(hash(9876543));  
    }  
    public static int hash(int number) {  
        return number % 100;  
    }  
}
```

## Array & LinkedList

Let's say hash table with 7 buckets (0, 1, 2, 3, 4, 5, 6)

Keys arrive in the Order (15, 11, 27, 8)



# HashTable

```
public class HashTable {  
    class Entry{  
        private int key;  
        private String value;  
  
        public Entry(int key,String value) {  
            this.key = key;  
            this.value = value;  
        }  
    }  
  
    private LinkedList<Entry>[] entries;  
  
    public HashTable() {  
        entries = new LinkedList[5];  
    }  
  
    @Override  
    public String toString() {  
        String s = "";  
        for (LinkedList <Entry> list: entries) {  
            if (list != null) {  
                for (Entry r: list) {  
                    s = s + " " + r.value + " ";  
                }  
                s = s + "\n";  
            }  
        }  
        return s;  
    }  
}
```

put

```
public void put(int key, String value) {  
    int index = hash(key);  
    if(entries[index] == null) entries[index] = new LinkedList<>();  
  
    var list = entries[index];  
  
    for(Entry entry: list) { // Check if key is already exist  
        if(entry.key == key) {  
            entry.value = value;  
            return;  
        }  
    }  
  
    Entry entry = new Entry(key, value);  
    list.addLast(entry);  
}  
  
private int hash(int key) {  
    return key % entries.length;  
}
```



get

```
public String get(int key) {  
    Entry entry = getEntry(key);  
    return entry != null ? entry.value : null;  
}  
  
private Entry getEntry(int key) {  
  
    int index = hash(key);  
    var list = entries[index];  
  
    if(list != null) {  
        for(Entry entry: list) {  
            if(entry.key == key) {  
                return entry;  
            }  
        }  
    }  
    return null;  
}
```

remove

```
public boolean remove(int key) {  
    int index = hash(key);  
  
    Entry entry = getEntry(key);  
    if(entry != null) {  
        entries[index].remove(entry);  
        return true;  
    }  
    return false;  
}
```

size

```
public int size() {  
    int size = 0;  
    for (LinkedList<Entry> list: entries) {  
        if (list != null) {  
            size = size + list.size();  
        }  
    }  
    return size;  
}
```

### Task 1

Berilgan String dagi birinchi takrorlanadigan harfni toping va bunda HashMap dan foydalaning.

Input: PDP Academy

Output: P

### Task 2

Berilgan String dagi birinchi takrorlanmaydigan harfni toping va bunda HashMap dan foydalaning.

Input: PDP Academy

Output: c

### Task 3

Berilgan String dagi eng ko`p takrorlangan harfni toping va bunda HashMap dan foydalaning.

Input: Amazing Academy

Output: a

### Task 4

Berilgan Array da farqi K bo`lgan juftliklar sonini topadigan dastur yozing va bunda HashMap dan foydalaning.

Input: arr[] = [1, 5, 3, 4, 2] k =3

Output: 2

{1,4} {5,2}

### Task 5

Berilgan Array da yig`indisi berilgan songa teng bo`lgan elementlarning index larini Console da chop qiladigan dastur yozing va bunda HashMap dan foydalaning.

Input: arr[] = [2, 6, 11, 17] target =8

Output: [0, 1]

2+6 = 8