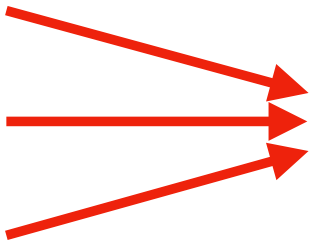
# HASHING DATA STRUCTURE

- **HASHING**
- **Search**
  **Insert**
  **Delete**



**O ( 1 )**

- **Key -> value**

**Phone Number Example** **-> If key is present then, overwrite the Value**

Hello world

# HASHING DATA STRUCTURE

- **HASHING  NOT useful**
- **1) Finding closet Value**
- **2) Sorted Data**

Hello world

# HASHING DATA STRUCTURE

- **HASHING Application**
- **1) Dictionaries**

**2) cryptography**

**3) caches**

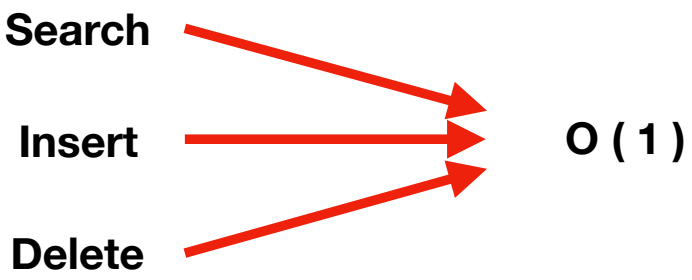
**4) Symbol Table in compilers/ Interpretors**

**5) Routers**

Hello world

# HASHING DATA STRUCTURE

- Direct Access Table
- Imagine a situation where you have 1000 keys with values from 0 to 999
  how would you implement in O (1 ) Time complexity




Hello world

# HASHING DATA STRUCTURE

- Direct Access Table
- Imagine a situation where you have 1000 keys with values from 0 to 999 how would you implement in O (1 ) Time complexity

**Example Situation**

insert(10)

insert(20)

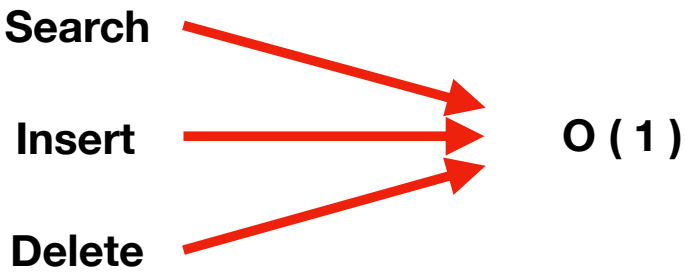
insert(119)

search(10)

search(20)

delete(199)

search(199)



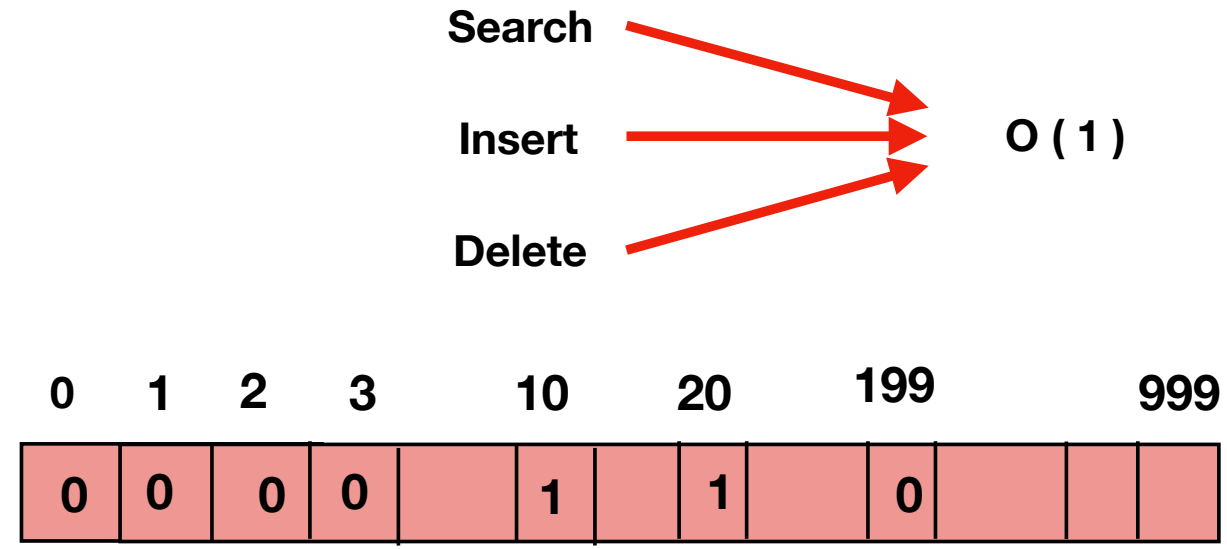

Hello world

# HASHING DATA STRUCTURE

- Direct Access Table
- Imagine a situation where you have 1000 keys with values from 0 to 999 how would you implement in O (1 ) Time complexity

**Example Situation**

insert(10)
insert(20)
insert(119)
search(10)
search(20)
delete(199)
search(199)




Hello world

# HASHING DATA STRUCTURE

- **Direct Access Table NOT works fine:-**

- **Phone numbers are key**

  **Floating , point numbers**

  **Keys are string**

**Because, it can't Handle Large Values**

Hello world