

# Hash Tables

Saurabh Mathur

VIT University

*14BIT0180*

September 29, 2015

# Overview

- 1 Why use Hashing
  - Word Count
  - Sets
  - Unique data representation
- 2 What is a Hash Table
  - Definition
  - Structure
- 3 Analysis
  - Trade-offs
  - Asymptotic Analysis

# Word Count

Read a text file, count the number of times each word appears, and print and how many times each word appears.

**Input:** A sequence of  $n$  words

**Output:** A list of (*word*, *frequency*) pairs

# Set

Read a text file and, print each unique word.

**Input:** A sequence of  $n$  words

**Output:** A list of  $k$  unique words ( $k < n$ )

# Unique data representation

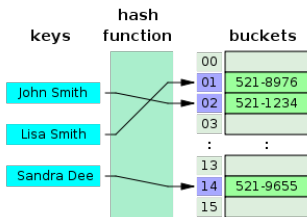
Read a sequence of text files and, print a string uniquely identifying each file.

**Input:** A sequence of  $n$  text files

**Output:** A list of  $n$  unique strings corresponding to each file.

# Definition

A **Hash Table** is a data structure used to implement the associative array ADT, a structure that can map keys to values.



# Structure

A hash table consists of -

- A Hash Function
- An array

# Hash Function

A hash function is a function that almost uniquely maps any value (called a key) to an index in the array. The key may be an integer, a string, a structure etc.

Eg. SHA-0, SHA-1, SHA-2, MD4, MD5



# Operations

```
void put(Key key, Value val)
Value get(Key key)
void remove(Key key)
```

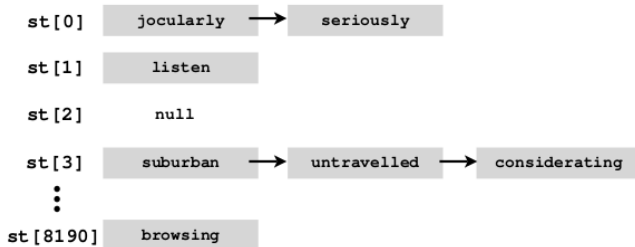
# Collision Resolution

- Probing
  - Linear
  - Quadratic
- Chaining

# Probing

<code>st[0]</code>	jocularly
<code>st[1]</code>	null
<code>st[2]</code>	listen
<code>st[3]</code>	suburban
<code>⋮</code>	null
<code>st[30001]</code>	browsing

# Chaining



# Trade-offs

- Memory sacrificed for speed and random access.
- No sequential/ordered access.

# Asymptotic Analysis

For an ideal hash, insertion, deletion and look-up are all constant time i.e.  $O(1)$

implementation	guarantee			average case			ordered iteration?
	search	insert	delete	search	insert	delete	
unordered array	N	N	N	N/2	N/2	N/2	no
ordered array	lg N	N	N	lg N	N/2	N/2	yes
unordered list	N	N	N	N/2	N	N/2	no
ordered list	N	N	N	N/2	N/2	N/2	yes

# The End