CENTER FOR TEXTUAL STUDIES AND
DIGITAL HUMANITIES

DIGH 401 - Introduction to Computing

Fall Semester 2014

Week 12

Today's Class

- Week 11 Exercise
- Data Structures
  - collections and dictionaries
  - hash tables
  - graphs
- Basic web design
  - HTML Part 1

Week 11 Exercise

- a simple check-in and check-out system for books (nothing too grandiose, simple push, pop, shift etc...)
  - Queue
- prioritised to-do list for a single user, which will allow them to add and delete to-do items in a specified order (the order is your choice...)
  - Stack or Queue (choice relative to required list functionality)

Data Structures - Queues

```
<?php

$students = array();

$students[0] = 'Emma';

echo $students[0].'<br />';

array_push($students, 'Catherine');

echo $students[1].'<br />';

echo array_shift($students);

?>
```

Data Structures - Collections and Dictionaries

Conceptual Intro

- use a collection or dictionary to store different data types together
- a collection acts like a resizable array
- a collection holds different data types
- a collection identifies each chunk of data with a number
- a dictionary acts like a collection that identifies each chunk of data with a unique key

Data Structures - Using a Collection

- acts like a super array that can grow and expand...
- a collection can store different data types, or even other data structures such as an array
- not all programming languages offer direct support for the collection data structure

Python
C or Pascal
C#, .Net etc
Java

Data Structures - Adding data to a Collection

- create a collection by declaring a variable as a collection

    MyStuff as New Collection

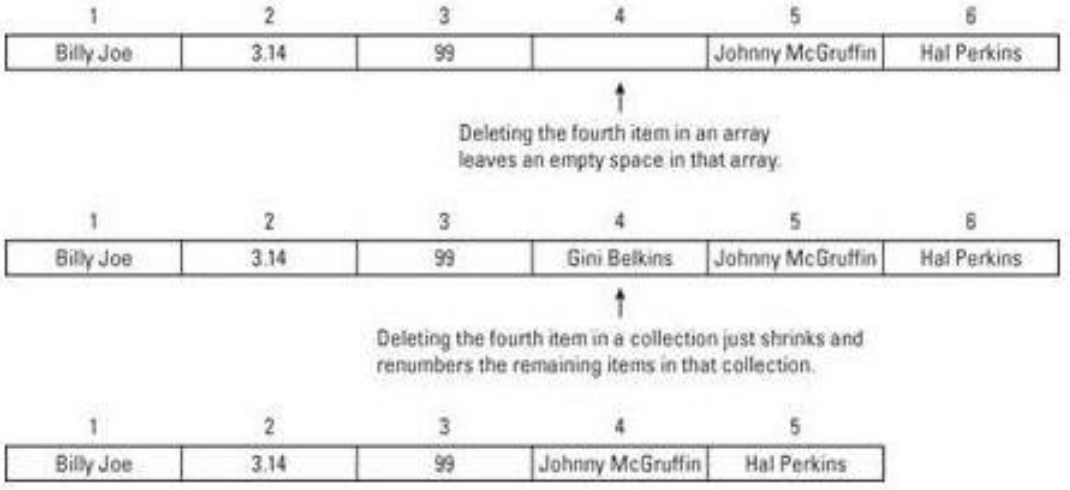
- when you create a collection it will contain zero items
- a collection will automatically expand when you add a new chunk of data
- each new element added to a collection is tacked onto the end
- every element in a collection gets numbered
- the first element is given a number of 1, then 2, and so on...
- you can also specify where to add new data in a collection using an index number

Data Structures - Deleting data from a Collection

- add data and delete data as necessary
- to delete data you must specify the location of that data by defining an index number

```
MyStuff.Remove(4);
```

- when you delete data from a collection, the collection automatically renumbers the rest of its data to avoid empty spaces



Example - unset()

Data Structures - Deleting data from a Collection

```
<?php

$students = array();

$students[0] = 'Emma';

echo $students[0].'<br />';

array_push($students, 'Catherine', 'Angela', 'Megan');

echo $students[1].'<br />';

unset($students[2]);

echo 'key 2 has now been unset from the array<br />';

print_r($students);

echo '<br />';

$i = 0;
$newStudents = array();
foreach ($students as $key => $value) {
      $newStudents[$i] = $value;
      $i++;
}

print_r($newStudents);

?>
```

Data Structures - Identification with keys

- collections identify data by their position in the collection
- if you don't know the specific location you will need to search the entire collection
- collections also give you the option to identify data with a descriptive string or 'key'
- allows you to use a descriptive string to help identify the data
- identify data by its given location or by a key
- criteria must be met when adding a key to data
    - the key must be added at the same time as the data
    - every key must be a string
    - every key must be unique

Associative Arrays in PHP

```
$students['teacher'] = 'Megan';
```

Example

Data Structures - Identification with keys

```
<?php

$students = array();

$students[0] = 'Emma';

echo $students[0].'<br />';

array_push($students, 'Catherine', 'Angela');

$students['teacher'] = 'Megan';

print_r($students);

echo '<br />Teacher = '.$students['teacher'];

?>
```

Data Structures - Searching and retrieving data from a Collection

- two direct ways to search and retrieve data from a collection
    - use the index number of the data
    - use the key of the data
- you need to keep a check of index numbers to be able to use that search option
- using index numbers to search is easy, but also unreliable
- keys are a good idea when we add an element
- use the key to find the required data
- you can always retrieve data from a collection with either its key or index no.

```
$students['teacher'];
```

Data Structures - Using a Dictionary

- similar to a collection but with two potential advantages
    - searching for data is much faster
    - key can be any type, including strings and numbers
- dictionaries use a 'hash tables' data structure
- keys must be used in a dictionary
- declare a variable for a dictionary

```
MyBook as New Dictionary;
```

Data Structures - Adding data to a Dictionary

- add data to a dictionary by defining both the data and the key to associate

```
MyBook.put(key, value);
```

- each time you add data to a dictionary you must add a corresponding key

Data Structures - Searching and retrieving data from a Dictionary

- identify the dictionary variable and key required
- use the key to find specific values

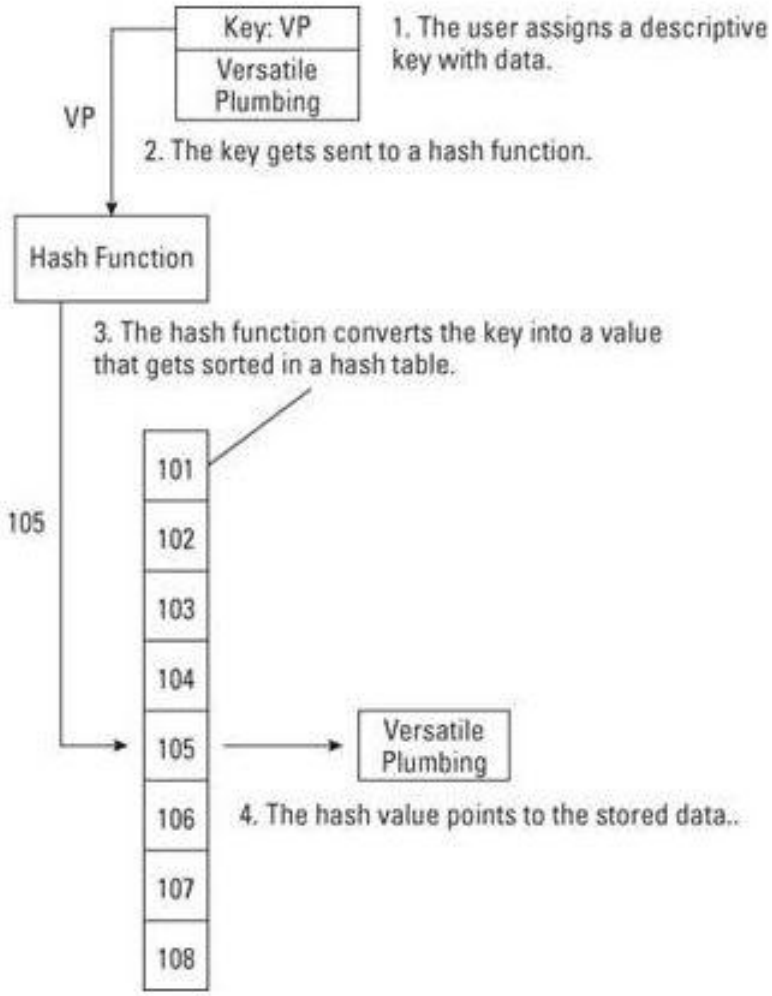
```
MyBook.get(key);
```

- searching a dictionary is more efficient because a computer does not need to search through the entire dictionary sequentially
- instead it searches through data using a hash table
- hash table makes searching faster by dividing data into distinct sections

Data Structures - Hash Tables

- dictionaries use hash tables to speed up searching
- a hash table takes the keys used to identify data and converts that key into a hash value
- hash value is stored in a sorted list

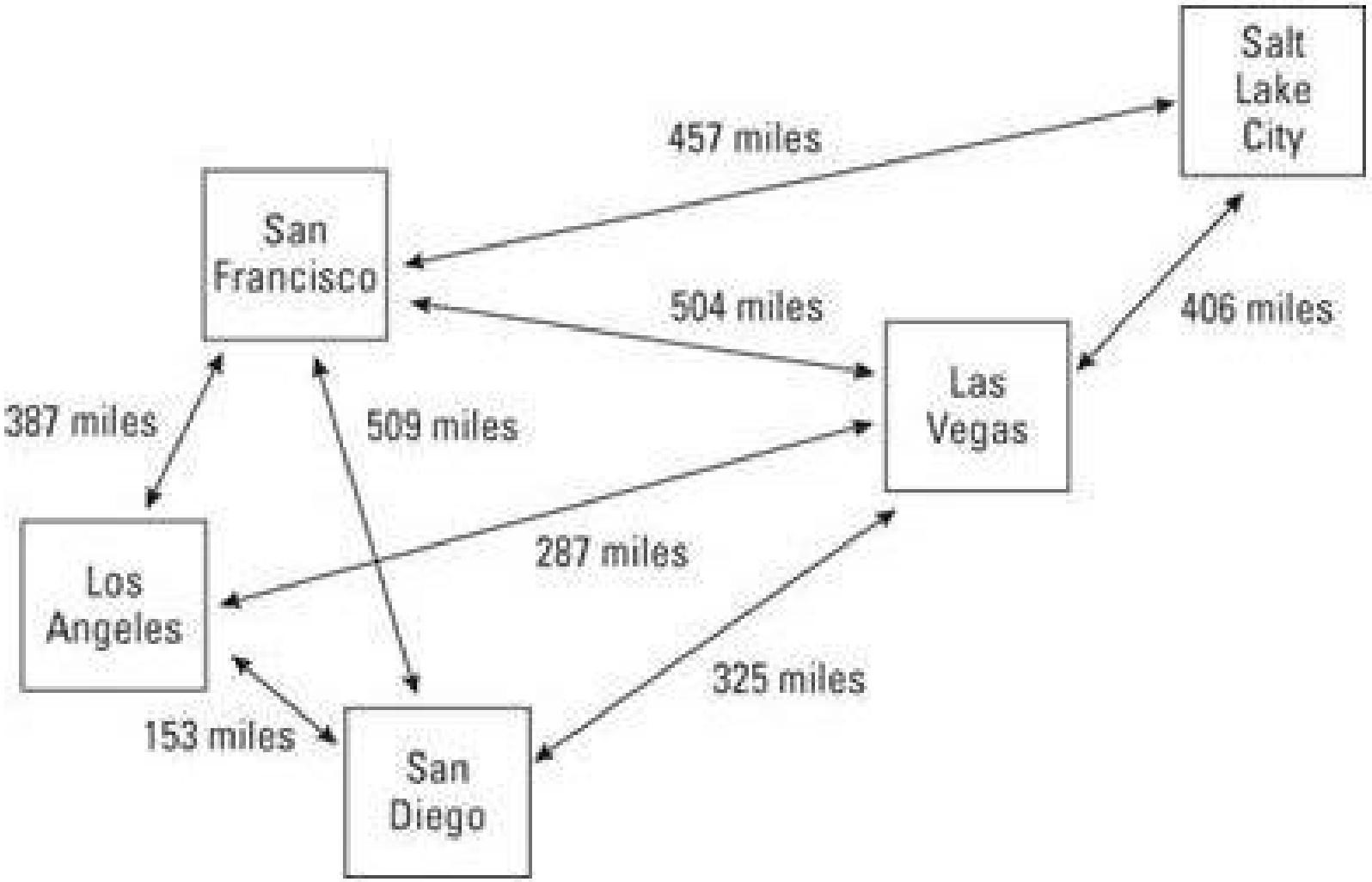
DIGH 401 - Introduction to Computing



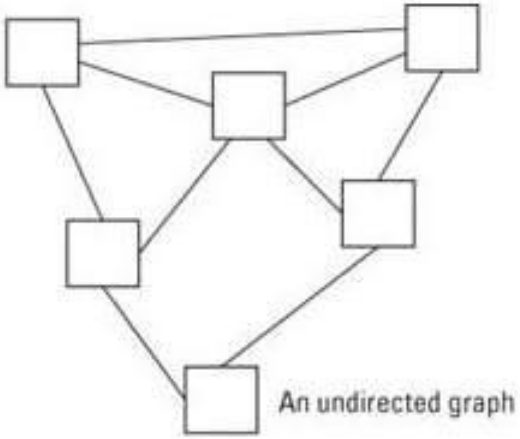
Data Structures - Retrieving data from a Hash Tables

- to retrieve data you give the computer the key associated with the required data
- a hash function is used to convert the key to a value
- this value is compared to a list of stored values in the hash table
- when a match is found the data associated with the key is returned

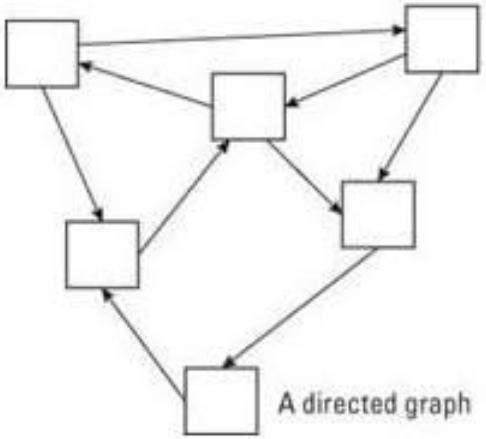
Data Structures - A brief look at Graphs



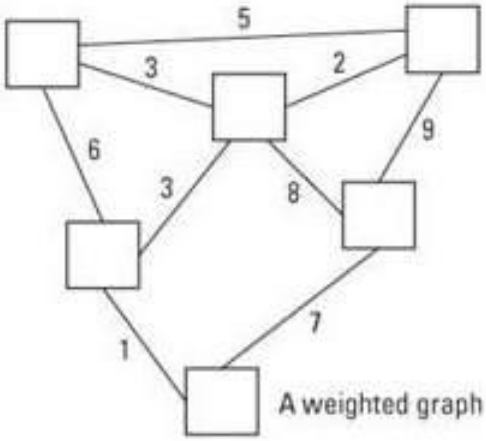
Data Structures - Different types of Graphs



An undirected graph



A directed graph



A weighted graph

Data Structures - Uses for Graphs

- often used to model a variety of real world problems
    - most efficient way to route email through a network
    - shortest route for a flight path...
- interesting hypothetical problems include
    - Seven Bridges of Königsberg
    - Chinese Postman Problem
    - Traveling Salesman Problem
    - Three Cottage Problem (Wikipedia / Math Forum)