CM1602 : Data Structures and Algorithms for AI

8. Maps, Sets, and Lists

Lecture 8 | R. Sivaraman

ROBERT GORDON UNIVERSITY ABERDEEN
TEF Gold
INFORMATICS INSTITUTE OF TECHNOLOGY

MODULE CONTENT

Lecture	Topic
Lecture 01	Introduction to Fundamentals of Algorithms
Lecture 02	Analysis of Algorithms
Lecture 03	Array and Linked Lists
Lecture 04	Stack
Lecture 05	Queue
Lecture 06	Searching algorithms and Sorting algorithms
Lecture 07	Trees
Lecture 08	Maps, Sets, and Lists
Lecture 09	Graph algorithms

Learning Outcomes

- LO1 : Describe the fundamental concepts of algorithms and data structures.
- On completion of this lecture, students are expected to be able to:
  - Describe Maps and when Maps are used.
  - Describe Sets and when Sets are used.
  - Describe Lists and when Lists are used.

Maps

Maps

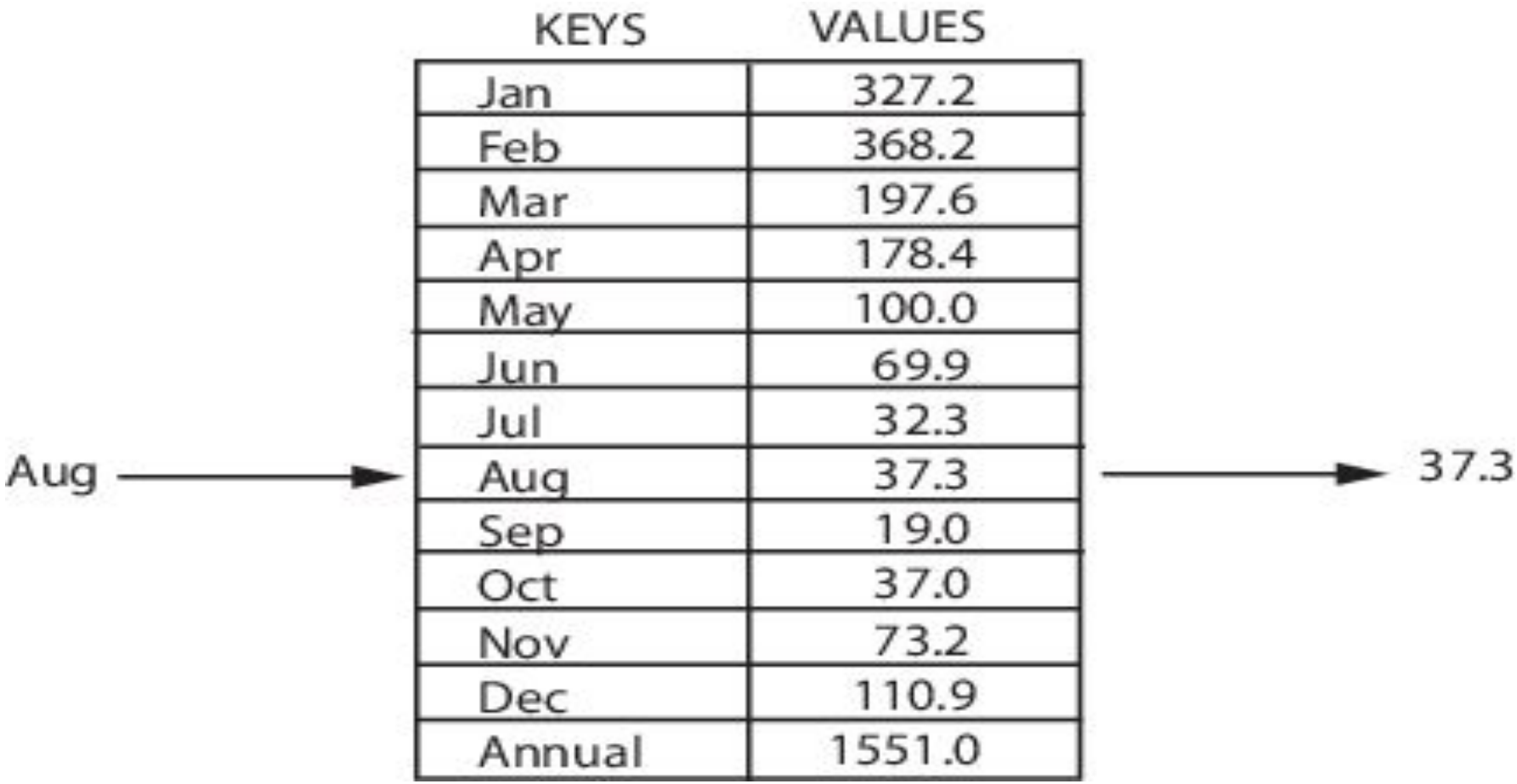
- A *Map* is a type of fast key lookup data structure that offers a flexible means of indexing into its individual elements.
- Also known as:
  - table, search table, dictionary, associative array, or associative container
- A data structure optimized for a very specific kind of search / access
  - with a *bag* we access by asking "is X present"
  - with a *list* we access by asking "give me item number X"
  - with a *queue* we access by asking "give me the item that has been in the collection the longest."
- In a *map* we access by asking "give me the *value* associated with this *key."*

Maps - Keys and Values

- Dictionary Analogy:
  - The *key* in a dictionary is a word:
    foo
  - The *value* in a dictionary is the definition:
    First on the standard list of metasyntactic variables used in syntax examples
- A key and its associated value form a pair that is stored in a map
- To retrieve a value the key for that value must be supplied
  - A List can be viewed as a Map with integer keys

Maps - Keys and Values

- Keys must be unique, meaning a given key can only represent one value
  - but one value may be represented by multiple keys
  - like synonyms in the dictionary.
    Example:
    factor: n.See coefficient of X
  - *factor* is a key associated with the same value (definition) as the key *coefficient of X*


KEYS
VALUES
Jan
327.2
Feb
368.2
Mar
197.6
Apr
178.4
May
100.0
Jun
69.9
Jul
32.3
Aug
37.3
Sep
19.0
Oct
37.0
Nov
73.2
Dec
110.9
Annual
1551.0
Aug
37.3

Sets

Sets

- A set is a collection of objects need not to be in particular order.
- It is just applying the mathematical concept in computer.
- Rules:
  - Elements should not be repeated.
  - Elements should have a reason to be in the set.

Sets

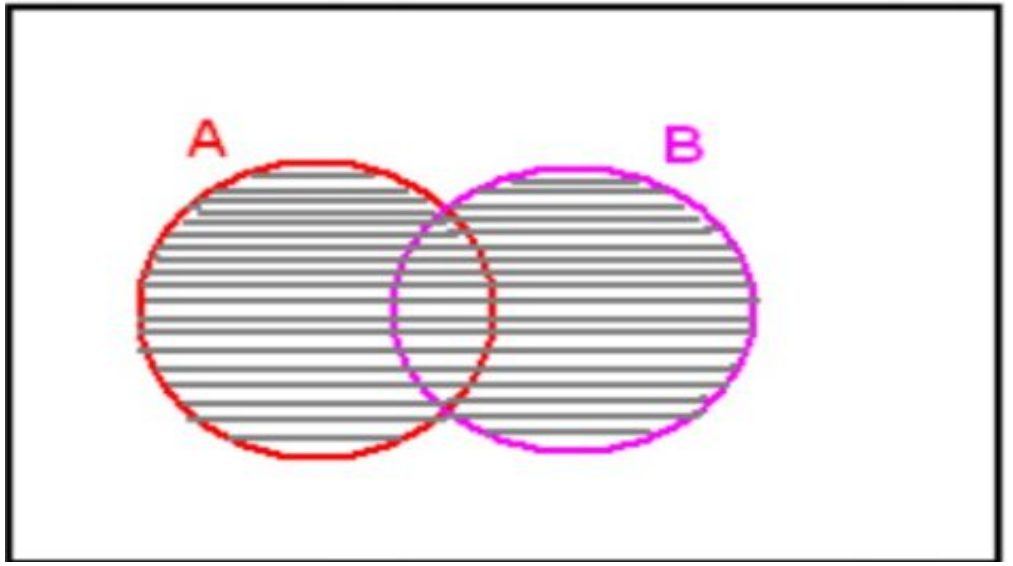
- Example: Assume we are going to store national cricketers in a set.
  - Names should not be repeated.
  - Name of a person not in the cricket team should not be there.
  - This is the restriction we found in the given set.

Sets

- The sets with no element is called a Null Set or Empty Set.
- Basic Set Operations:
  - set union (set1, set2)
  - set intersection (set1, set2)
  - set difference (set1, set2)
  - set subset (set1, set2)

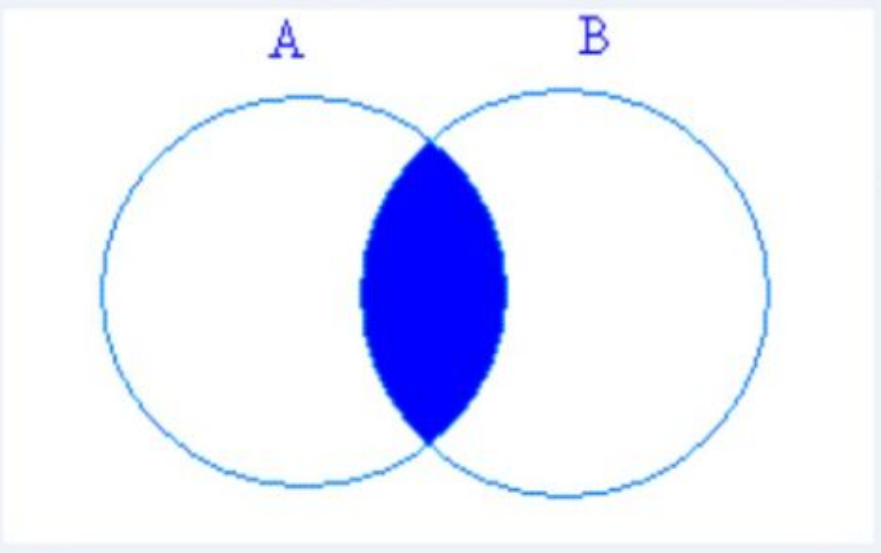
Sets - Union

- Combine two or more sets without violating the rules for set.



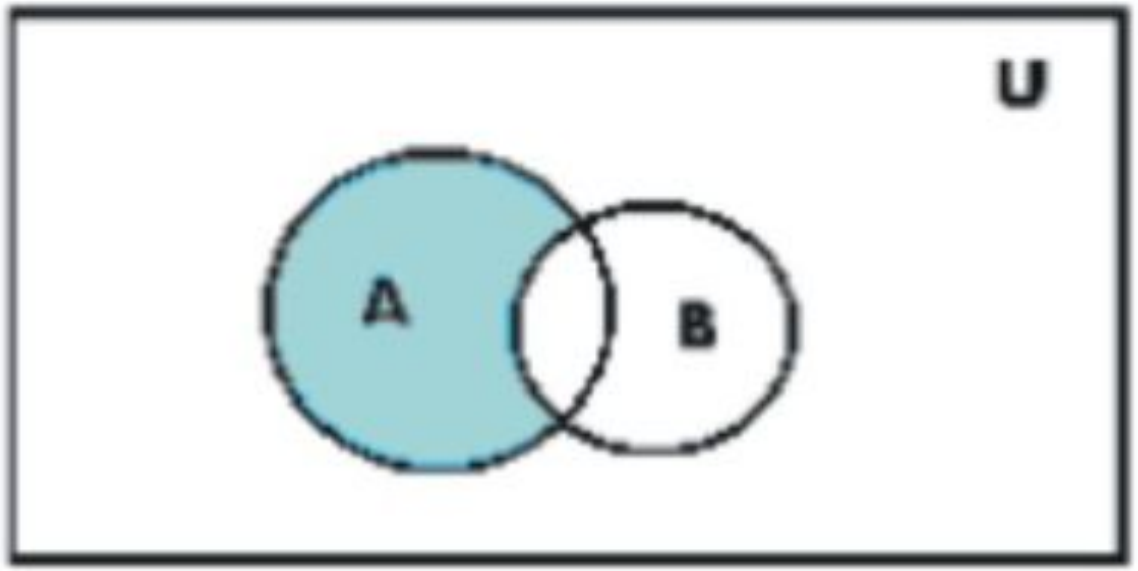
Sets - Intersection

- Gathering common elements in both the sets together as a single set.



Sets - Difference

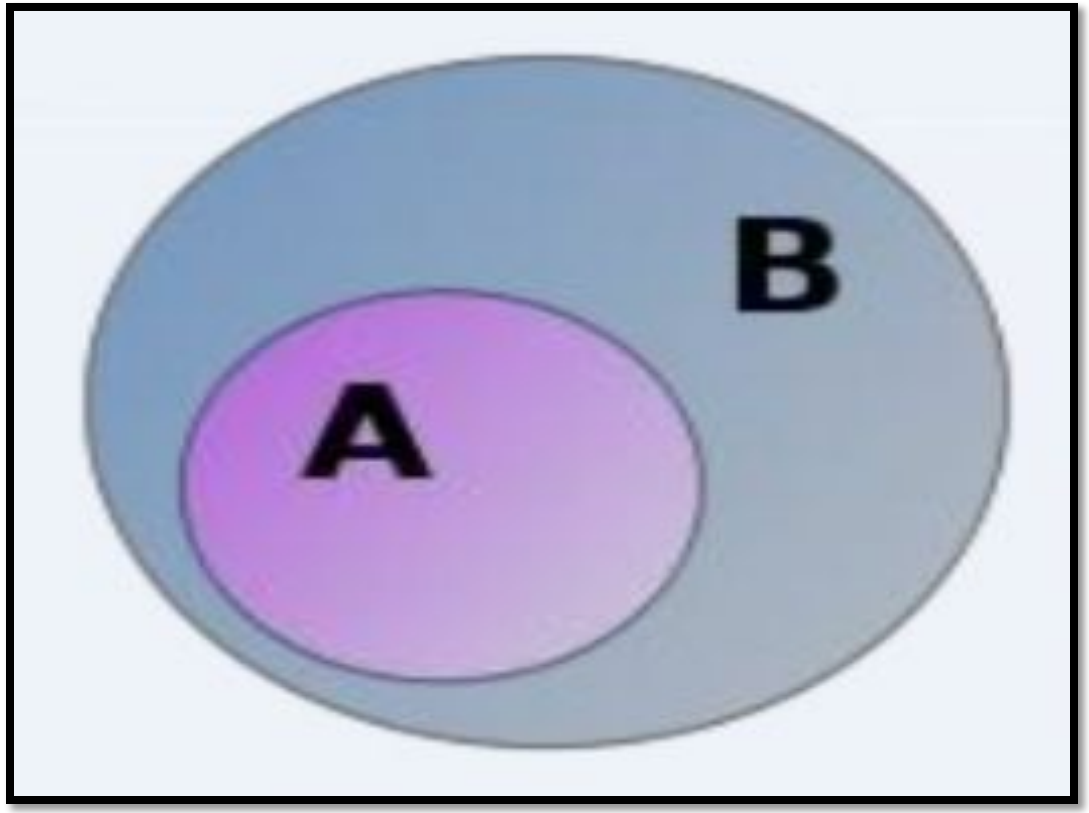
- Forming a set with elements which are in first set but not in second set.



A - B

Sets - Subset

- A is a subset of B. i.e. A contained in B.



Lists

Lists

- List defines a sequential set of elements to which you can add new elements and remove or change existing ones.
- The list data structure typically has two very distinctive implementations — array list and linked list.
- Array List: It is basically a self-resizing array or, in other words, a dynamic array.
- Linked List: Refer Lecture 3.