Lecture - 01
Data Structures

Data Structure

Data Structure

The logical and mathematical model of a particular organization of data is called a data structure.

A **data structure** is a particular way of storing and organizing data in a computer so that it can be used efficiently.

common data structures:

- Arrays,
- Linked lists,
- Stacks,
- Queues,
- Trees,
- Graph

Types of Data Structure

There are two types of data structure

Linear Data Structure

A data structure is said to be linear if its elements form a sequence, or in other words a linear list.

- Array
- Stack
- Queue
- Linked List

Non- Linear Data Structure

A non-linear structure is mainly used to represent data containing a hierarchical relationship between elements.

- Tree
- Graph

Array

Linear array (One dimensional array) : A list of finite number n of similar data elements referenced respectively by a set of n consecutive numbers, usually 1, 2, 3,…..n. That is a specific element is accessed by an index.

Let, Array name is A then the elements of A is : a_1,a_2….. a_n

Or by the bracket notation A[1], A[2], A[3],…………., A[n]

The number k in A[k] is called a subscript and A[k] is called a subscripted variable.

- Homogeneous collection of values (all the same type)
- Store values sequentially in memory
- Associated INDEX with each value
- Use array name and index to quickly access 'K' th element

Example

A linear array STUDENT consisting of the name of six students

STUDENT_ID

1	15
2	9
3	18
4	22
5	10
6	35

Here, STUDENT_ID[4] denote 22

Array (con…)

Linear arrays are called one dimensional arrays because each element in such an array is referenced by one subscript.

(Two dimensional array) : Two dimensional array is a collection of similar data elements where each element is referenced by two subscripts.

Such arrays are called matrices in mathematics and tables in business applications.

Multidimensional arrays are defined analogously

MATRICES

	1	2	3	4
1	1	2	3	4
2	5	6	7	8
3	9	10	11	12
4	13	14	15	16

Here, MATRICES[2][3]= 7

Array Data Structure

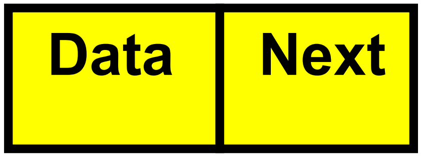
- It can hold multiple values of a single type.
- Elements are referenced by the array name and an *ordinal* index.
- Each element is a *value*
- Indexing begins at *zero.*
- The array forms a *continuous* list in memory.
- We specify the array size at compile time, often with a named constant.

Linked lists

- A linked list, or one way list, is a linear collection of data elements, called nodes, where the linear order is given by means of pointers.

- Dynamically allocate space for each element as needed.

Node



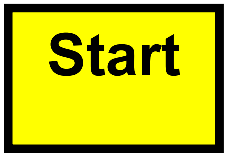
In linked list

Each node of the list contains the data item

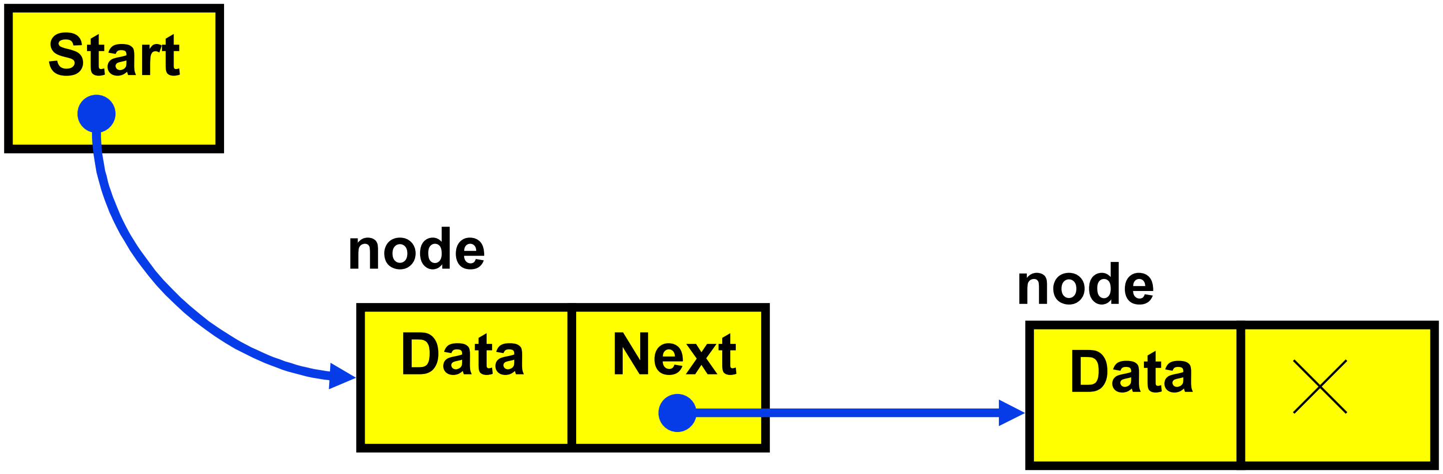
a pointer to the next node

Collection structure has a pointer to the list Start

Initially NULL

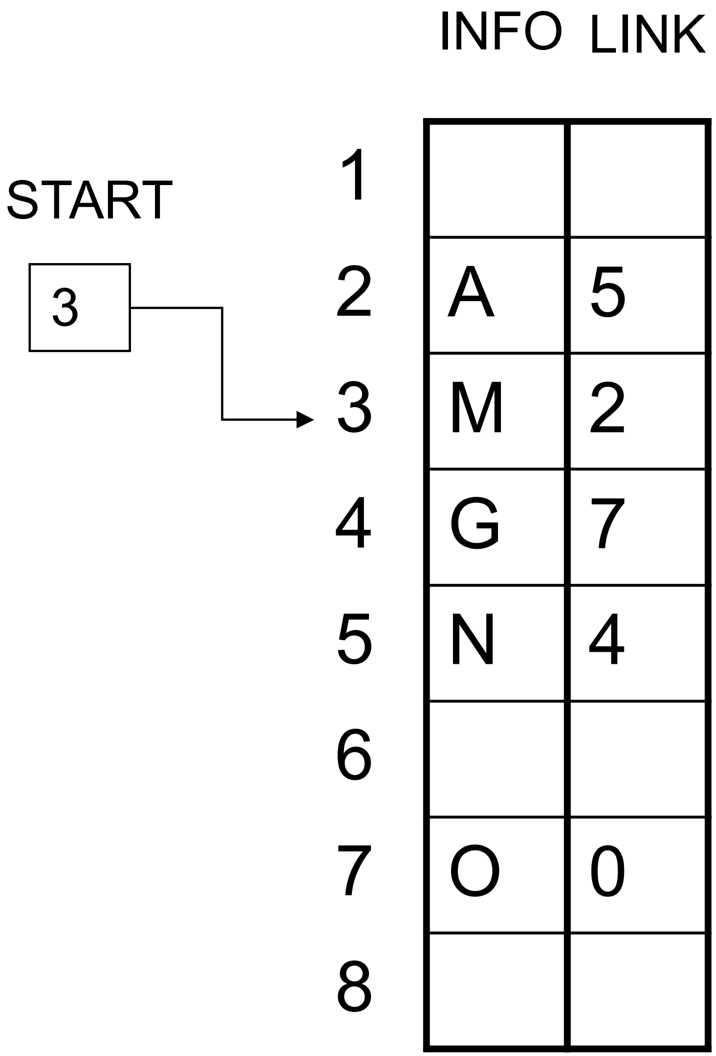


Linked lists



Linked list with 2 nodes

Linked lists



	INFO	LINK
1		
2	A	5
3	M	2
4	G	7
5	N	4
6		
7	O	0
8		

START=3, INFO[3]=M

LINK[3]=2, INFO[2]=A

LINK[2]=5, INFO[5]=N

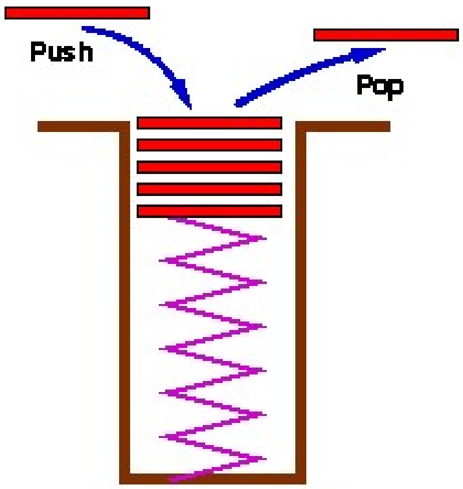
LINK[5]=4, INFO[4]=G

LINK[4]=7, INFO[7]=O

LINK[7]=0, NULL value, So the list has ended

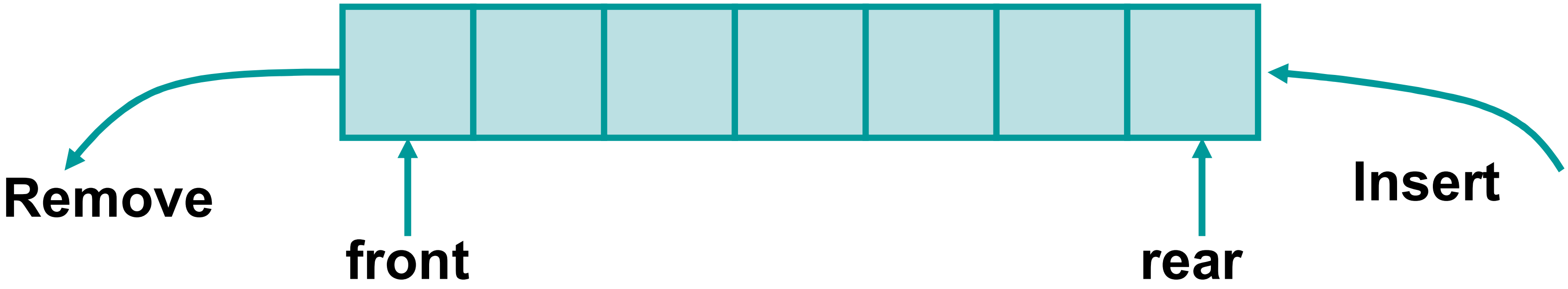
Stacks

- Stacks are a special form of collection with LIFO semantics
- Two methods
  - add item to the top of the stack
  - remove an item from the top of the stack
- Like a plate stacker

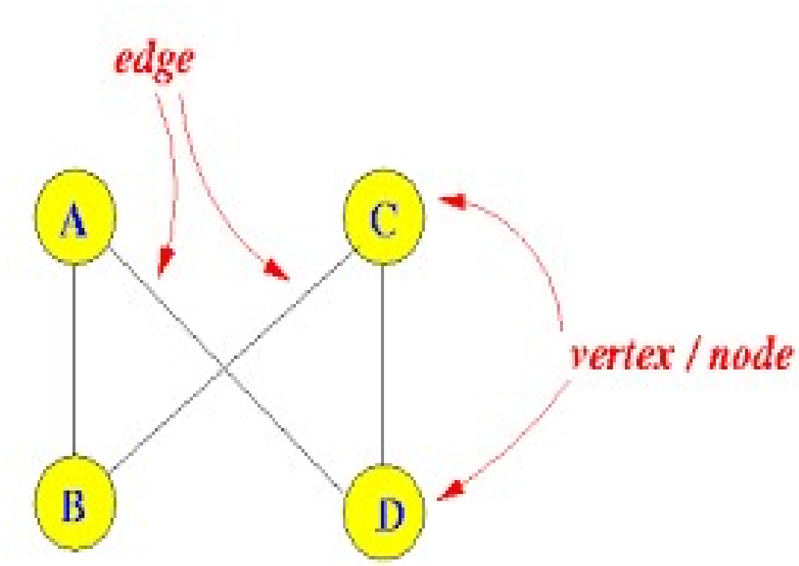


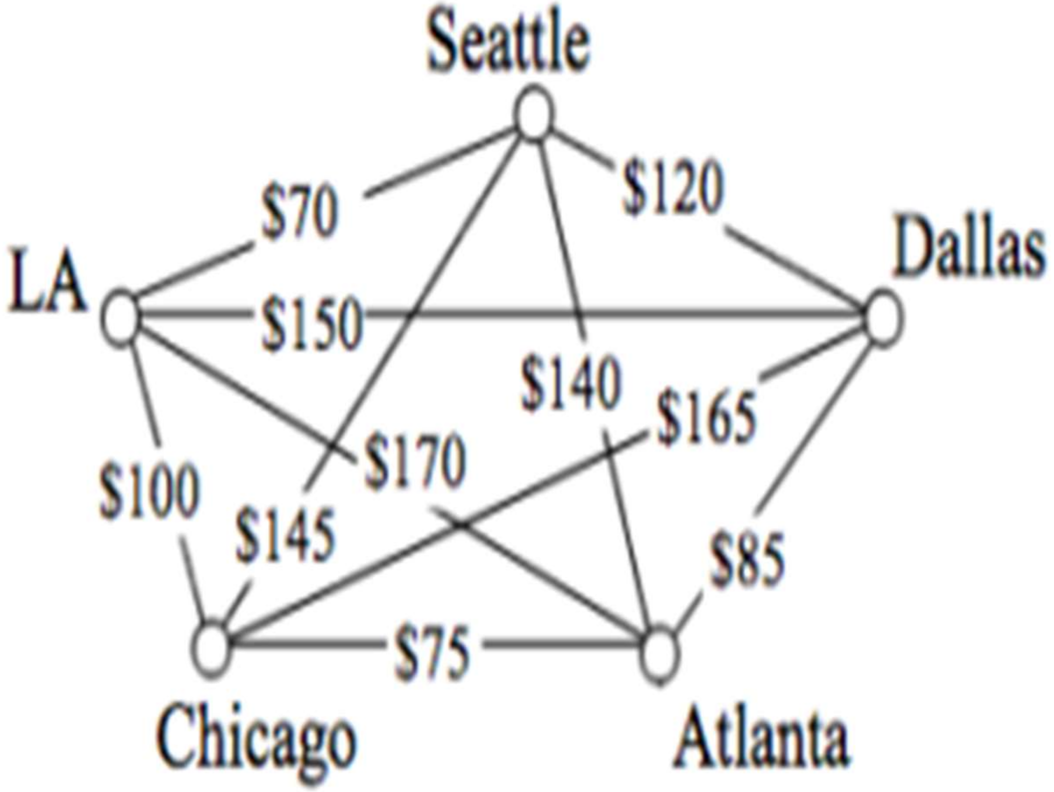
Queues

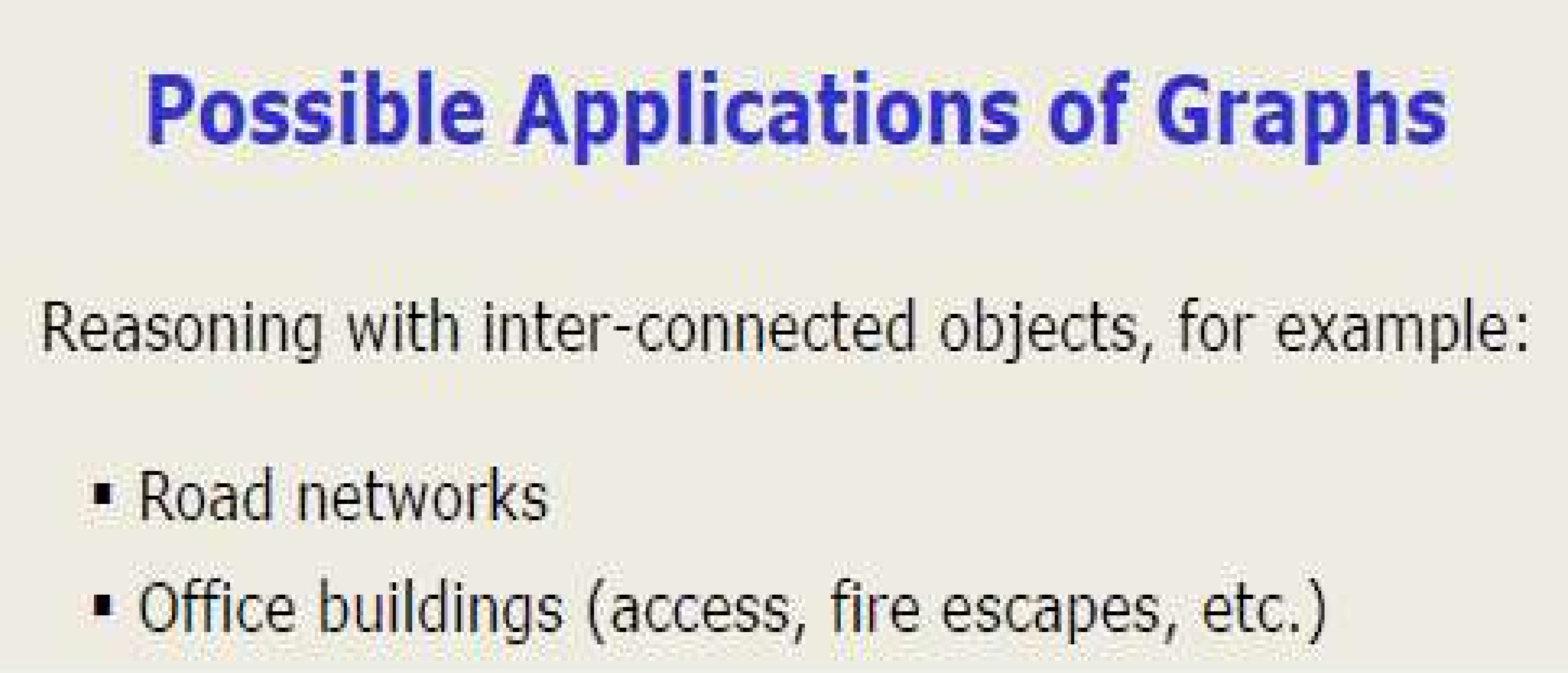
- Like a stack, a *queue* is also a list. However, with a queue, insertion is done at one end, while deletion is performed at the other end
- The insertion end is called *rear*
  - – The deletion end is called *front*



Graph


edge
A
C
B
D
vertex / node


Seattle
$70
$120
LA
Dallas
$150
$140
$165
$170
$100
$145
$85
$75
Chicago
Atlanta



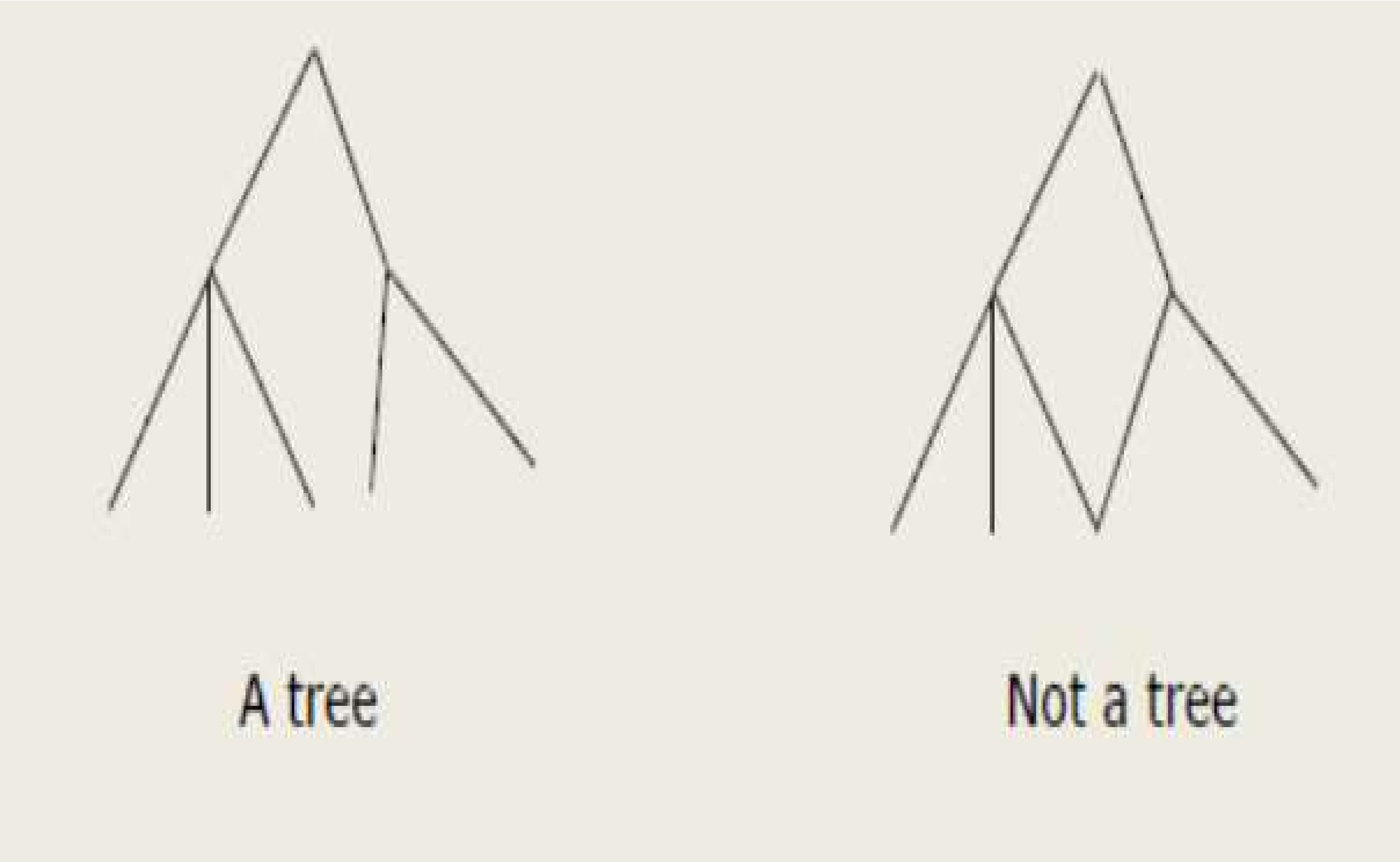
Possible Applications of Graphs

Reasoning with inter-connected objects, for example:

- Road networks
- Office buildings (access, fire escapes, etc.)

Tree

- A tree is a graph that does not contain a cycle.



Data structure operations

The data appearing in our data structure is processed by means of certain operations. The following four operations play a major role:

Traversing

Accessing each record exactly once so that certain items in the record may be processed. (This accessing or processing is sometimes called 'visiting" the records.)

Searching

Finding the location of the record with a given key value, or finding the locations of all records, which satisfy one or more conditions.

Inserting

Adding new records to the structure.

Deleting

Removing a record from the structure.

Data structure operations (Continued)

The following two operations, which are used in special situations, will also be considered:

Sorting:

Arranging the records in some logical order

Merging:

Combining the records in two different sorted files into a single sorted files

Thank You