Dated: 19-Aug-2024

Data Structures: logical/mathematical management of data & Algorithms

Timespace Tradeoff → Balance between memory (space) and time taken to execute by a data structure.

(strategic decision to balance between time efficiency & memory usage based on requirements of an application)

→ optimizing one another is often inversely proportional.

e.g:

For web-based application memory is not an issue as the website will operate on live server or cloud. But response (time to retrieve) is an important requirement.

For mobile application response may not be a issue but memory must be taken to consideration and in most cases, should not exceed 1gb or 2gb.

e.g: (Real life example)

High-speed sports car consume more fuel but deliver greater speeds whereas economical family cars run on moderate speeds but offer better fuel consumption. Attaining higher speed comes at the cost of more fuel

5 Basic operations that must be available by any kind of Data structure:

i. Traversing: accessing the data atleast once
ii. Insertion: adding new data to the structure
iii. Deletion: deleting specific data without hindering the rest
iv. Edit: updating or modifying the data once after it was set.
v. Search: being able to search through the structure for a specific data.

2 Additional operations:

i. Merging: combining the records in two different data structures.
ii. Sorting: arranging records in some logical order

Dated:

* Data Structure: → logical or mathematical model of particular organization of data.
→ way of organizing a collection of data

- Records: files
- Node: Linked lists, trees & graphs
→ an individual part of a larger data structure
- Data representation

Linear:
- Stack
- Queue
- Array
- Linked list (logically)

Non-linear:
- Trees
- Graphs
- Linked list (physically)

- Algorithms

→ systematic instructions on a task
→ stepwise

Use of:
→ sequence
→ selection
→ iteration
→ recursion

- 2D Array:

int arr[5][2]
↓ ↓
rows colours

- 3D Array:

int arr3[2][5][2]
↓ ↓ ↓
pages rows columns

Dated:

Pseudocodes

Algorithm Headers
- → name
- → parameters & types
- → purpose
- → precondition (precursor requirements for the parameters)
- → post-condition (taken action and status of parameters)
- → return condition (return value)

Algorithm Body
- → statements
- → statement numbers
- → variables
- → algorithm analysis
- → statement constructs (sequence, selection, iterati

* Rule of three: (→ for exception-safe code during copies and deletion/destruction)

Define all three when a class defines atleast once of these:

→ Destructor → Copy constructor → Copy Assignment operator

* Jagged Array:
  - → array of arrays
  - → multidimentional array
  - → different number of columns, changed dynamically
  - → may take up a big chunk of continuous memory on the heap

arr[4][ ]
- arr[0][3]
- arr[1][2]
- arr[2][5]
- arr[3][1]

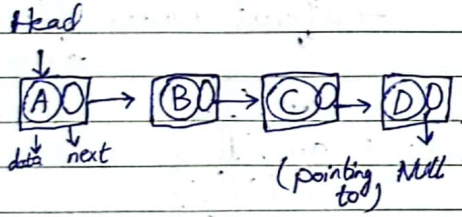
Dated: 26-Aug-2024

* Rule of three: explicity define
  → Destructors → Copy Constructors - Assignment Operator
to get rid of dangling pointers upon deletion of objects)

* Issue with shallow copy:
  → causes objects to be initialized on the same stack.
  → if one object is deleted, then other copies in the stack are also deleted.
⇒ Can be solved by Deep Copy: a new object of same contents is created that exists in the memory as a completely separate entity.

•Linked List:

→ collection of data and pointers
→ Physically non-linear whereas logically linear structure.
→ e.g: structure or class containing:
  • data of any type
  • node pointer pointing to the next node.




```
class node{
  public:
    node(): next(NULL){}
int data;
node *next; };
```

Types:
- Singly linked - Double linked
- Circularly linked
- Mixed

Dated:

* Functionalities of linked list:
  i. Access → Traversing/Display
  ii. Edit/Append
  iii. Searching → linear
                 → Interpolation
  iv. Insertion
  v. Deletion
  vi. Merge
  vii. Sorting
    - Bubble
    - Selection
    - Insertion
    - Radix
    - Shell
    - Comb