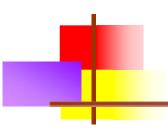


What is Data Structure?

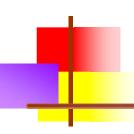


Data Structures

■ **Data:** Any digital file stored in computer system, such as: audio/video file, word file, pdf file, images, text file, etc.

Data structures is a way to organize, process, retrieve, store data efficiently.

For specific purposes, appropriate data structures can be used.



Dictionary: sorted data

- Data is words with their meanings.
- Words are arranged in Alphabetic order (sorted Data)
- Searching is fast
- Makes it easier to insert new words at proper place

abomination | 2

bad **abominable snowman** large apelike creature said to live in the Himalayas **abominably** *adv*

abomination *n* someone or something that is detestable

aborigine [ab-or-rij-in-ee] n 1 original inhabitant of a country or region 2 **Aborigine** sometimes offens indigenous Australian aboriginal adj

abort v 1 have an abortion or perform an abortion on 2 have a miscarriage 3 end a plan or process before completion abortive adjunsuccessful

abortion n 1 operation to end a pregnancy 2 informal something grotesque abortionist n person who performs abortions, esp. illegally

abound v be plentiful **abounding** adj

about prep 1 concerning, on the subject of 2 in or near (a place) ▷ adv 3 nearly, approximately 4 nearby about to shortly going to not about to determined not to about-turn n complete change of attitude

above adv, prep 1 over or higher (than)
2 greater (than) 3 superior (to) above
board in the open, without
dishonesty

abracadabra *n* supposedly magic word

abrasion n scraped area on the skin abrasive adj 1 harsh and unpleasant in manner 2 tending to rub or scrape > n 3 substance for cleaning or polishing by rubbing

abreast *adv*, *adj* side by side **abreast** of up to date with

abridge v shorten by using fewer words **abridgment** or **abridgement**

abroad *adv* 1 to or in a foreign country 2 at large

abrogate v cancel (a law or agreement) formally abrogation n abrupt adj 1 sudden, unexpected 2 blunt and rude abruptly adv abruptness n

abs pl n informal abdominal muscles **abscess** n inflamed swelling containing pus

SPELLING TIP

There is a silent c in the middle of

abscess that's easy to forget

abscond v leave secretly

abseil [ab-sale] v go down a steep drop by a rope fastened at the top and tied around one's body

absent adj 1 not present
2 lacking 3 inattentive ▷ v 4 stay
away absently adv absence n
1 being away 2 lack absentee n
person who should be present but
is not absenteeism n persistent
absence from work or school absentminded adj inattentive or
forgetful absent-mindedly adv

absinthe n strong green aniseed-flavoured liqueur

absolute adj 1 complete, perfect 2 not limited, unconditional 3 pure: absolute alcohol absolutely adv 1 completely ▷ interj 2 certainly, yes absolutism n government by a ruler with unrestricted power

absolve v declare to be free from blame or sin **absolution** n

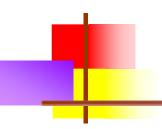
absorb v 1 soak up (a liquid)
2 take in 3 engage the interest of
(someone) absorption n absorbent
adj able to absorb liquid absorbency
n

abstain v 1 choose not to do something 2 choose not to vote abstainer n abstention n abstaining, esp. from voting abstinence n abstaining, esp. from drinking alcohol abstinent adj

abstemious adj taking very little
alcohol or food abstemiousness n
abstract adj 1 existing as a quality

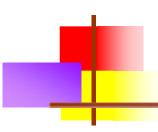
abstract *adj* 1 existing as a quality or idea rather than a material object

a

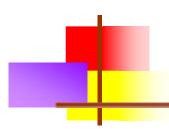


Data Structures are Classified into Two Types

- Linear data structure: arrange sequentially or linearly where each and every element is attached to its previous and next adjacent.
- Array
- Linked list
- Stacks
- Queues
- Non-linear data structure: not arranged sequentially or linearly
- Trees
- Graphs



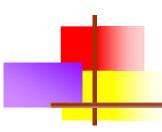
Array Data Structure



- Suppose we are looking for a specific data value from 50,000 items
- [dbh nik aks rby cjk maf yyt qmn bbn pkm fad rsa udi]

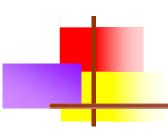
■ The data could be in beginning of the array or towards the end...

It would be lot easier if data were organized.

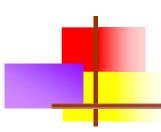


- One can carry out different operations on array elements
 - **Search** for an element
 - **Delete** an element
 - **Insert** a new element

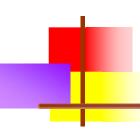
- *Deletion* and *Insertion* would need lot of operations
- Searching would need very few operations if array is sorted



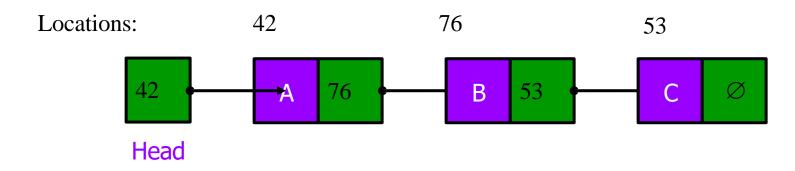
Is it possible to handle the problem without shifting so many data values?



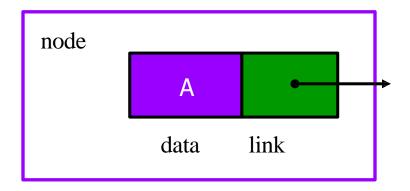
Linked List Data Structure

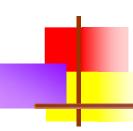


Linked List Data Structure

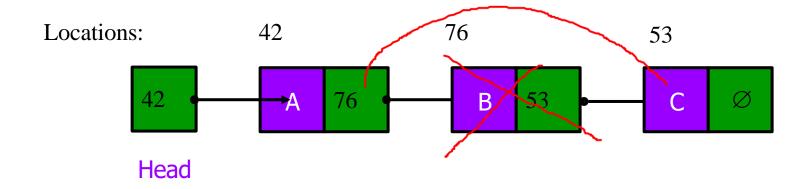


- A linked list is a series of connected nodes
- Each node contains
 - Data (any type)
 - Link to the next node in the list
- The last node points to **NULL**

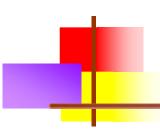




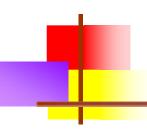
Operations on Linked List



- Deletion
- Insertion
- Searching



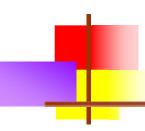
Queue Data Structure



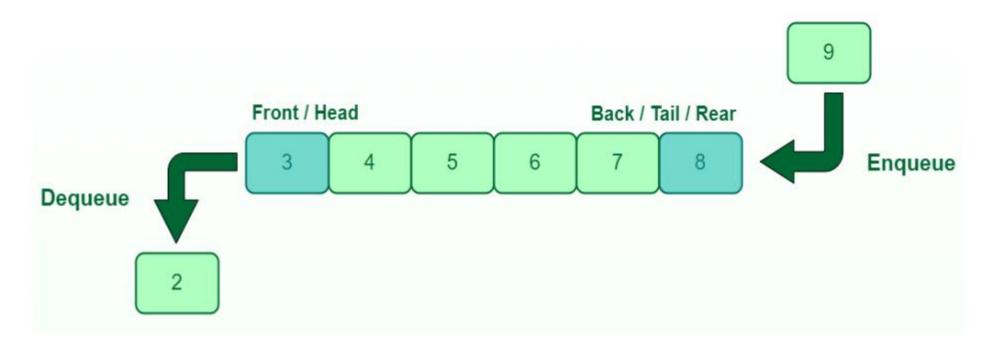
Queue Data Structure

- Cannot join in any random position in the list
- Must join at end of the list
- Only the first person in the list gets serviced (front of the list),
 for ex: bank waiting line

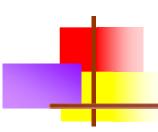




Queue Data Structure



Queue Data Structure



Stack Data structure

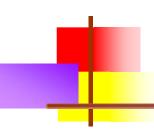


Stack Data Structure

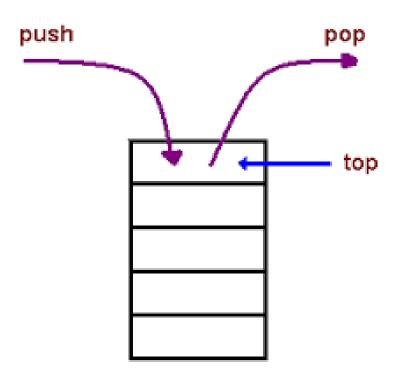


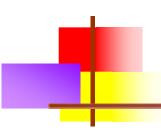




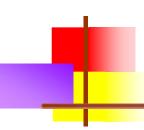


- You cannot access any random element of stack
- Only top element can be accessed or removed

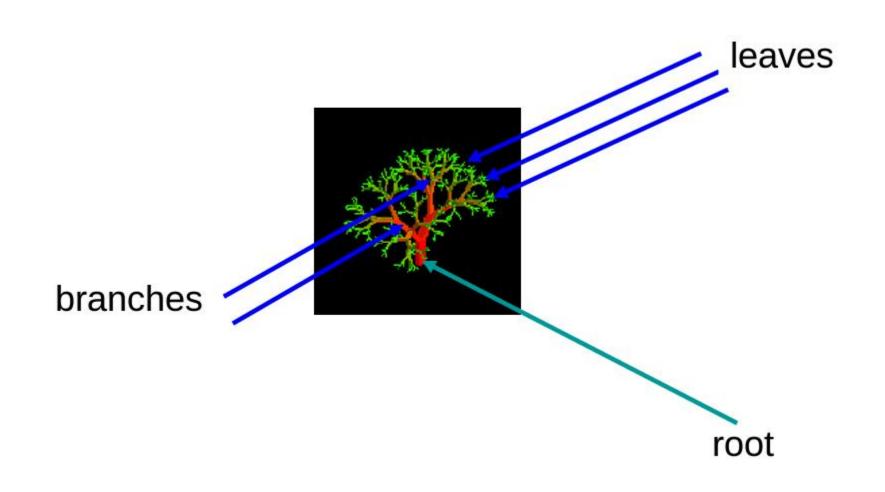


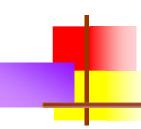


Tree

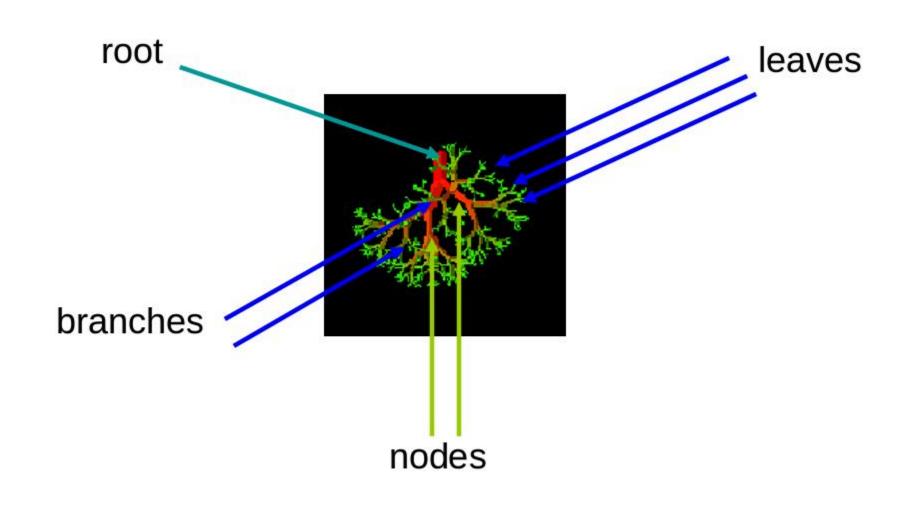


Nature Lover's View of A Tree





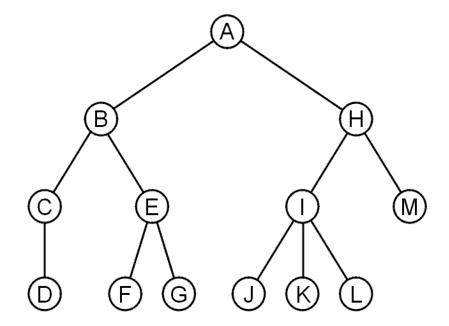
Computer Scientist's View

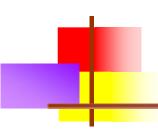




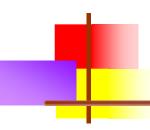
A tree data structure stores information in *nodes*

- Similar to linked lists:
 - There is a first node or **root**
 - Each node has a variable number of **references** to successors
 - Each node, other than the root, has exactly one node pointing to it



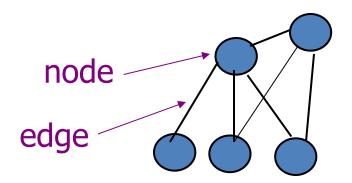


Graph



What is a graph?

- Graphs represent the relationships among data items
- A graph G consists of
 - a set V of **nodes** (vertices)
 - a set E of edges: each edge connects two nodes
- Each node represents an item
- Each edge represents the relationship between two items



Your project report should be a concise document (1-2 pages) with the following structure:

- 1. Briefly describe the purpose of your project and its primary functionality. Also highlight the main functions to achieve this functionality in 1 or 2 paragraph.
- 2. In the **next paragraph**, explain how you have applied concepts from ICP to enhance the efficiency of your code such as dynamic memory allocation, file handling, or other relevant features from the course.
- 3. This is an **optional paragraph**, describing any unique features, optimizations, or innovative logic implemented in your project.

ICP Marks & Assessment Discussion

Type of Assessment	Description	Percentage (%)
Assignments (Best 8 out of 9)	Total 9 was taken (Assignment 1 was no need to submit)	15
Graded labs (Best 2 out of 3)	Taken 3 graded labs	10
Graded lab 4: Project	Project with report	5
Quizzes (Best 2 out of 3)	Subjective +Objective	10
Surprise Quizzes (Best 5 out of 6)	Total 6 has been taken (even 8)	5
Midterm Exam	Subjective + Objective	20
Final Exam	Subjective + Objective	35
	Total	100

- Today is the last ICP lecture.
- If you have query in the marks obtained in any of the evaluated graded component such as Assignments, Mid-sem, Quizzes, Graded labs or any query on project then feel free to meet me on **Friday 1 to 3.**

End-sem Exam Format

- Total number of Questions: ~15 (Objective, theory, and Programming questions);
- Marks ~60 Marks
- Duration 2 hours to 2hours 15 minutes
- Syllabus:
 - ~ 25 to 30% from before mid-sem exam (Topics: Introduction to Basic Fundamentals of Computers, Introduction to Programming, Identifiers and Constants, Data Types, Number System, Operators, Logical Expressions, Managing input & output, Conditional statements, Decision making & Branching, Decision making & loops, Arrays, Functions)
 - ~ 70 to 75% from after mid-sem exam (Topics: Functions, recursion, pointers, Searching and sorting, String, Structure & Union, bit manipulation, file handling, Intro to Data Structure).

End-sem Exam Format Cont...

- Types of Questio:
- 1. 5 MCQ questions (you need to justify your answers in 1 line): 10 Marks.
- 2. 5 Output types questions (Code snippet would be given to you): 20 marks
- **3. 3 theory questions** (may include difference or explain any topic): 15 marks
- 4. 2 Programming questions: 15 marks

MCQ Question Sample

#include <stdio.h> int main() { int *p = NULL; if (p) { printf("Pointer is not null\n"); } else { printf("Pointer is null\n"); return 0; } What would be the output of this code? a. Pointer is not null b. Pointer is null

d. Run time error

c. Compilation error

Ans: b; Here, if statement is checking for whether P is not NULL

Output Type Question

```
#include <stdio.h>
  int main() {
    int a = 10;
    int b = 20;
    int *p = &a;
    printf("%d\n", *p);
    p = \&b;
    printf("%d\n", *p);
    return 0;
Output: 10
           20
```



Output Type Question

```
#include <stdio.h>
int main() {
  int x = 10, y = 20;
  int *const p = &x;
  printf("%d\n", *p);
  *p = y;
  printf("%d\n", *p);
  return 0;
}
```

Output: 10

20

Reason:

p must always point to the same memory address (in this case, the address of x). However, the value of the memory address can still be modified.



Theory Questions

- Difference between Various Searching techniques.
- Difference between Structure and Union
- Explain strcmp() and strcat() with syntax



Programming Questions

Programming Questions



Thank you & All the best