

ORAL QUESTION FOR DSA/DSL

2-TTU

(Data Structure) UNIT-1

- 1) Define Data structure, Data & Object?
- 2) Define Algorithm & Flowchart, Problem?
- 3) Characteristics of Algorithm and operation.
- 4) List steps in Problem Solving.
- 5) Problem solving techniques.
- 6) Symbol used in flowchart.
- 7) e.g. of Algorithm & Flowchart.
- 8) Different Approaches to design algorithm.
- 9) Difference bet Top-down vs. Bottom approach.
- 10) Cases of analysis of Algo.
- 11) Time complexity & space complexity.
- 12) Define Asymptotic notation & its type.
- 13) List some common asymptotic notation.
- 14) Need of data structure, ADT?
- 15) Types of data structure.
- 16) Compare Linear vs. Non-linear Data structure.
- 17) State Algorithm strategies.
- 18) Divide & Conquer strategy with e.g.
- 19) Greedy strategies with e.g.
- 20) What is recurrence relation & its types?

- 8) Advantages & Disadvantages of array.
- 9) Applications of array.
- 10) Single variable polynomial, operation, Polynomial representation using array.
- 11) State sparse matrix & its representation with e.g.
- 12) Transpose of sparse & its representation with e.g.
- 13) Fast transpose matrix with e.g.
- 14) String manipulation using array.
- 15) Write prog. to reverse a string.
- 16) Write prog. to transpose of matrix.

UNIT-3

Linked List

- 1) What is Linked list & its representation.
- 2) Advantages/Disadvantages of linked list.
- 3) Application of linked list & its terminology.
- 4) Difference bet Array vs. Linked list.
- 5) Linked list Operation/ADT.
- 6) Dynamic memory management.
- 7) Difference bet malloc() & calloc() fun.
- 8) Types of Linked list: (SLL, DLL, CLL)
- 9) Compare Singly vs. Doubly vs. Circular LL.
- 10) Polynomial using Linked list. Perform Addition & multiplication of LL.
- 11) State Generalized linked list representation.
- 12) GLL e.g. 1) {3, {4, 5, 3}, 6, {{7, 8, 9}, 10, 3}}
- 2) (a, b, f, c, d, f(e, f), h), (f))
- 3) (a(b, c, d), e, f)
- 4) ((a, (b, c)), d, e) etc.
- 5) (P, Q, (r, S(u, v), w), (x, y))
- 12) Graphical collection of Advantages/Disadvantages

UNIT-2

Sequential Organization

- 1) Definition Sequential organization.
- 2) Define Array, declare array.
- 3) Array Representation, operation of array.
- 4) Formula for 1D or 2D Address calculation.
example of 1-D or 2D Address calculation
- 5) ADT for Array. write code to reverse array.
- 6) Explain 2-D array (Def, declarⁿ, represen^t)
- 2) State row major & col major representation

UNIT-4

STACK

- 1) Defⁿ of stack - represⁿ of stack - operations
- 2) stack as an ADT: Component of stack.
- 3) multiple stack. Applications of stack.
- 4) write prog. to reverse string using stack.
- 5) what is Polish notation. state its types.
- 6) list operators & its priority in notation.
- 7) write algo infix to postfix conversion
 - i) $A * B + C * D$
 - ii) $((A+B) * (C-D)) / E$
 - iii) $a \uparrow b * c - d + e / f / (g+h)$
 - iv) $((a / (b-c+d)) * (e-a) * c)$
 - v) $A * (B + C * D) + E$
- 8) write algo postfix to infix conversion
 - i) abcde $\wedge \wedge * - [] /$
 - ii) abc--de-fg-h+/*
- 9) write algo postfix to prefix conversion
 - i) abcde $\wedge \wedge * -$
- 10) write algo postfix evaluation: expr^s
 - i) $53+82-*$
 - ii) $562+*124/-$
 - iii) $623+-382++*2^3+$
 - iv) $532*8+*$
 - v) $636+5*9/-$
- 11) write algo infix to prefix expression
 - i) $(A-B/C) * (D * E-F)$
 - ii) $((A+B) * (C-D)) / (E+F)$
- 12) write algo Prefix into Postfix conversion
 - i) $* + a - bc / - de + - fgh$
- 13) stack operation using linked list
- 14) Recursion using stack - represⁿ
- 15) Advantages/Disadvantages of recursion
- 16) Backtracking Algo strategies
- Define 8-queens Problem.

UNIT-5

QUEUE

- 1) Queue, represⁿ, operation of queue.
- 2) Queue as an ADT.
- 3) Compare stack vs Queue.
- 4) operation of Queue using Linked list.
- 5) Circular queue & its operation (program)
- 6) Dequeue & its operation (Program)
- 7) Priority queue & its ADT.
- 8) Applications of Queue.
- 9) $Front = 2$; $Rear = 3$ (linear queue)

Queue = $\rightarrow A, D, \rightarrow, \rightarrow, \rightarrow$

 - i) Add 'S'
 - ii) Add 'J'
 - iii) Delete two letters
 - iv) shift towards left to bring all free space to right side.
 - v) Insert M, H, L & delete ~~one~~ letter.
- 10) Circular queue

Front = 2; Rear = 4

Queue = $\rightarrow, A, C, D, \rightarrow, \rightarrow$

 - i) F is added
 - ii) two letter are deleted
 - iii) R is added
 - iv) S is added
 - v) One letter is deleted
- 11) A linear queue using array has a size of 3.
 - i) Insert 10
 - ii) Insert 20
 - iii) Insert 30
 - iv) delete element
 - v) Insert 40
 - vi) delete element
 - vii) delete element
 - viii) delete element
 - ix) Insert 50

UNIT-6

1) Def searching and sorting. its types

2) state types of searching tech.

3) Define Linear search & Binary search.

4) Algo. for Linear / Binary search.

5) Compare Linear Vs Binary search.

6) Binary search example.

i) 20, 35, 37, 40, 45, 50, 51, 55, 67

key = 37

ii) 5, 9, 11, 15, 25, 29, 30, 35, 40.

key = 29

7) state sentinel search & fibonacci search.

8) write Algo. for fibonacci search.

9) Sorting, types of sorting techniques

10) Define stability of sorting

11) explain Bubble sort Algo. of Bubble sort

eg. i) 6, 5, 9, 6, 2, 8, 1.

ii) 10, 5, 4, 18, 17, 1, 2.

iii) 9, 7, -2, 4, 5, 3, -6, 2, 1, 8.

iv) 10, 9, 8, 7, 6, 5, 4, 3, 2, 1.

12) Insertion Sort: How it works? Algorithm

i) 5, 0, 1, 9, 2, 6, 4.

ii) 20, 10, 8, 6, 4, 2, 1

iii) 1, 7, 3, 2, 0.

iv) 50, 10, 78, 40, 30, 02, 04, 15.

13) Selection Sort: its Algorithm

eg. i) 5, 9, 1, 11, 2, 4.

ii) 50, 20, 70, 40, 30

iii) 10, 6, 13, 7, 5, 51, 27, 2, 3, 15, -3, 4.

iv) 29, 72, 98, 13, 87, 66, 52, 51, 36.

v) 76, 67, 36, 55, 23, 14, 6.

GROUP A

14) Quick Sort Algorithm

eg. 1) 20, 54, 48, 37, 12, 92, 86, 07.

2) 54, 26, 93, 17, 77, 31, 44, 55, 20.

3) 6, 4, 5, 8, 2, 3, 1, 9, 5.

4) 4, 2, 6, 5, 3, 9.

5) 30, 35, 10, 15, 20, 34, 5, 18, 6, 11, 13, 26, 39.

15) Merge sort Algorithm

eg. 1) 24, 11, 9, 2, 6, 5, 4, 3.

2) 56, 12, 84, 56, 28, 0, -13, 47, 94, 31.

3) 20, 24, 48, 37, 12, 92, 86, 07.

4) 50, 10, -10, 40, 15, 25, 20, 35, 30.

16) Shell sort Algorithm

eg. 1) 5, 1, 9, 8, 2, 4, 6, 19.

2) 8, 3, 2, 11, 5, 14, 0, 9, 4, 20.

3) 8, 3, 2, 11, 5, 14, 0, 2, 9, 4, 20.

17) Bucket sort Algorithm / Radix algo

eg. 1) 121, 235, 55, 973, 327, 179.

2) 10, 5, 99, 105, 55, 100, 135, 141, 137, 200, 199.

3) 14, 1, 66, 74, 22, 36, 41, 59, 64, 54.

18) Heap Sort: Algorithms: open & Heap.

eg. 1) 18, 13, 12, 22, 15, 24, 10, 16, 19, 14, 30.

2) 8, 3, 2, 11, 5, 14, 0, 2, 9, 4, 20.

19) Comparison of All sorting Complexity.

(Techniques in brief, time complexity, stability)

20) Write code for Bubble sort.

2) Time complexity of All & Practical.

GROUP-A.

(Student)

PRACTICAL NO: 01

- 1) write ^{Algo.} ~~code~~ to store marks of PSA subject
 - i) Avg score of class
 - ii) Highest score / lowest score of class
 - iii) Marks scored by most of student
 - iv) list absent student for test.

- 2) What is concept of array with e.g.?
- 3) How to run c/c++ code in ubuntu? ^{linux}
- 4) What is long form of GCC/g++?
- 5) What is use of /a.out cmd?
- 6) Why we use using namespace std?
- 7) What is use of # symbol in c++?

- 8) What is long form of iostream?
- 9) What is syntax of for loop/while loop?
- 10) What is meaning of >> & << operator?
- 11) Time Complexity of program?
- 12) Who is inventor of C & C++ lang?
- 13) When C/C++ lang. is developed?
- 14) What is diff. bet C/C++?
- 15) Why C lang. called POP?
- 16) Why C++ lang. called OOP?
- 17) What are features of OOP/C++ lang?
- 18) State Data Encapsulation?

- 19) What do you mean of cout & cin?
- 20) Diff. bet angular < > & <> in header file.
- 21) Diff. bet class vs object.
- 22) Diff. bet = & == operator.
- 23) When void keyword used in function?
- 24) What is postfix increment & prefix increment in C/C++ lang?
- 25) Access specifier in C++?

2-TIME

PRACTICAL NO: 02 (password)

- 1) What is permutation?
- 2) How to generate password?
- 3) What is use of '10' charact.?
- 4) What are types of passing argument?
- 5) Diff. bet Actual & formal parameter.
- 6) What is nested loop?
- 7) What is use of * this symbol?
- 8) Write algo to generate password?
- 9) What is time complexity of this prog?

PRACTICAL NO: 03 (matrix)

- 1) Write algo for this prog.
- 2) State its time complexity.
- 3) Perform matrix addition, multiplication, transpose of matrix, upper triangular or lower triangular matrix.
- 4) What do you mean by 2-D array?
- 5) What is use of break; keyword?
- 6) What is use of endl in C++?
- 7) Write logic for summation of diagonal ^{value}?
- 8) What is ^{value} storage size of int (max size)?

PRACTICAL NO: 04 (string)

- 1) Write algo for string operation.
- 2) State its time complexity.
- 3) What are inbuilt string functions?
- 4) Write string function (operation) without using inbuilt function.
- 5) Function to convert string to upper letter / lower letter.
- 6) What is use of gets() function?

Prepared By - Prof. Anand Ghosh

(pinacle club) GROUP - BPRACTICAL NO-05

- 1) what is linked list? it's types representation, applications.
- 2) write algo for singly linked list.
- 3) state it's time complexity.
- 4) How to declare structure (struct)?
- 5) What is pointer in C++? *
- 6) How to declare class? syntax of it?
- 7) diff. bet' struct and class in C++?
- 8) Describe function() by passing value?
- 9) what is use of :: scope resol. symbol?
- 10) write syntax for do-while loop?
- 11) write logic to reverse linked list?
- 12) what is use of new & delete keyword?
- 13) what is malloc() & calloc() for?
- 14) what's use of sizeof() function?
- 15) what are diff. ways to insert or delete element in linked list.
- 16) Diff bet' Array vs linked list.

(cinemax ticket booking)

PRACTICAL NO-06

- 1) what doubly circular linked list. represent, advantages, application?
- 2) write algo for this program.
- 3) state it's time complexity.
- 4) what is use of constructor & destructor?
- 5) How to declare constructor in C++?
- 6) How to assign memory to node.
- 7) write logic to book/cancel ticket?
- 8) How to call function using object.
- 9) Diff bet' reference vs pointer.
- 10)

PRACTICAL NO-07

- 1) what is SLL? repres? advantages application?
- 2) write algo for this prog.
- 3) state it's complexity.
- 4) How to declare node for SLL?
- 5) write syntax for switch() case.
- 6) How to allocate/deallocate mem in SLL?
- 7) state logic to book/cancel appointment.
- 8) state logic for swapping.

GROUP - C

(well formed parenthesis)

PRACTICAL NO-08

- 1) what is stack, representation, Advantages application.
- 2) Compare stack vs linked list/Array.
- 3) what is well formed parenthesis.
- 4) write algo for this program.
- 5) state it's time complexity.
- 6) what is use of #define?
- 7) what are operation of stack.
- 8) How to push/pop element from stack?
- 9) what are cond for stack empty & full?

(infix to postfix)

PRACTICAL NO-09

- 1) what are polish notation.
- 2) write algo for infix to postfix conversion.
- 3) list operator as per precedence.
- 4) write complexity of this program.
- 5) use of isalpha() & isdigit() function.
- 6) what are rule for operator precedence.

Prepared By = Prof. Anand Ghosh

GROUP-D

PRACTICAL NO-10 (Job queue)

- 1) What is Queue. its represⁿ, advantages, applications.
- 2) stack vs Queue (Difference)
- 3) use of front & rear.
- 4) Types of Queue?
- 5) write algo for Queue operation?
- 6) state its time complexity.
- 7) can you delete node from rear?
- 8) can you add node from front?
- 9) what are cond^s for empty & full queue.

PRACTICAL NO-11 (Pizza Order)

- 1) What is circular queue, its representaⁿ, advantages, applications.
- 2) Diff. betⁿ Simple & Circular queue?
- 3) what is formula for Circular queue to insert/delete element.
- 4) Write algo for Circular queue operation?
- 5) state its time complexity.
- 6) What is use of friend class?
- 7) what are cond^s for insert() & delete() queue.

PRACTICAL NO-12 (Deque)

- 1) What is doubly ended queue, its represⁿ, advantages, Application.
- 2) what are operation of deque.
- 3) write algo for deque operation.
- 4) state its complexity.
- 5) what is ternary operator?
- 6) Compare queue vs Dequeue.

GROUP-E (selection/)

PRACTICAL NO-13 (bubble)

- 1) What is bubble & selection sort?
- 2) solve e.g. of bubble & selection sort.
- 3) write algo. for bubble & selection sort.
- 4) state time complexity of sorting.
- 5) Compare diff. sorting algo.
- 6) what is logic to display top 5 scorer. It.
- 7) How many comparison required for bubble & selection sort.
- 8) How many passes are required.
- 9) what are the issues of sorting.

PRACTICAL NO-14 (quick sort)

- 1) What is quick sort, representaⁿ, advantages, applications.
- 2) write algo for quick sort.
- 3) state its time complexity.
- 4) How many ways to select pivot.
- 5) which algo strategy used in quick sort.
- 6) solve e.g. of quick sort.

PRACTICAL NO-15 (shell & Insertion)

- 1) What is shell & Insertion sort.
- 2) solve e.g. of shell & Insertion sort.
- 3) write algo for shell & insertion sort.
- 4) state time complexity of algo.
- 5) Explain logic of shell sort.
- 6) passes needed for shell sort.
- 7) passes needed for insertion sort.

*** BEST OF LUCK ***

Prepared By - Prof. Anand N. Ghami.