PYQ - 3rd Sem (DSH) 2021-2022. Date: 19-10-23

Group-A

1. Answer the following:

(i) What is the time complexity of an infix to postfix conversion algorithm?

Ans - O(N) or O(n)

(ii) Which of the following is essential for converting an infix expression to postfix notation

Operator Stack.

(iii) A data structure is a particular way of organizing and storing data either in computer's memory or on the disk storage so that it can be used efficiently.

(iv) Quick Sort is not a stable sorting algorithm

(v) Time complexity of bubble sort in best case is

Ans - O(n)

(vi) When a sorting technique is called stable?

Ans - It keeps elements with equal keys in the same relative order in the output as they were in the input then the technique is called stable

(vii) What is the search complexity in direct addressing?

Ans - O(1)

(viii) What is best case for linear search?

Ans - O(1)

(ix) Reverse Polish notation is often known as

Ans- Postfix Notation.

(x) The value in a BST can be sorted in ascending order by using which of the traversal method?

Ans- Inorder Traversal

(xi) Which notation comprises a set of all functions h(n) that are greater than or equal to cg(n) for all values of $n \geq n0$?

Ans- Asymptotic notation

(xii) A circular queue is implemented using an array of size 10. The array index starts with 0. Front is 6, and rear is 9. The insertion of next element takes place at the array index

Ans- 0.

Group-B

Ques2- Write an algorithm for binary search technique.

Ans- Step 1: Initialize the search space to be the entire sorted array.

Step 2: Take the middle element of the search space and compare it to the target value.

Step 3: If the target equals the middle element you have found the target value and can terminate the search.

Step 4: If the target is less than the middle element, then the left half of the search space is used for the next search.

Step 5: If the target is greater than the middle element, then the right half of the search space is used for the next search.

Step 6: Repeat steps 2-5 until the target is found or the entire search space has been exhausted.

3Ques- State the differences between stack and queue data structure

Ans-

Stack	Queue
• It represent the collection of element in last in first out (LIFO) order.	• It represents the collection of elements in first out (FIFO) order
• Objects are inserted removed at the same end called Top of stack (TOS).	• Object are inserted and removed from different ends called front and rear ends.
• Insert operation is called push operation.	• Insert operation is called Enqueue operation
• In stack there is no wastage of memory space	• Delete operation is called Dequeue operation
• Plate counter at marriage reception is example of Stack.	• In Queue, there is a wastage of memory space.
• Delete operation is called Pop operation.	• Ex- line at movie theatre.

Ques 4 - Convert the following ... into postfix expression using the stack data structure with detailed explaination :—

$A - (B/C + (D \% E * F) / G) * H$

Ans-

Ques 5 - Show how the following polynomial can be represented using a linked list.

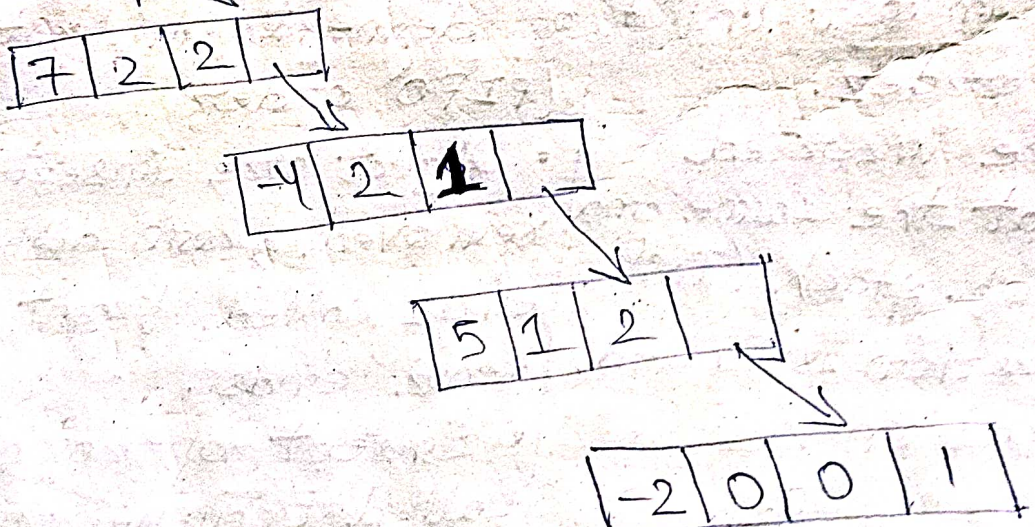
$7x^2y^2 - 4x^2y + 5xy^2 - 2$

Ans- Representation of Polynomial using Linked List.

Each node will have four parts :—

Coefficient	Power of 'x'	Power of 'y'	Next address

The polynomial is



Ques 6 - Write difference b/w b tree and b+ tree

ns-

B-tree	B+ Tree
ll internal and leaf odes have data pointers.	• Only leaf nodes have data pointers.
nsertion takes more time d it is not predictable sometime	• Insertion is easier and the results are always the same.

- Leaf nodes are not stored as structural linked list.
- No duplicate of keys is maintained in the tree
- For a particular number nodes height is larger.

- Leaf nodes are stored as structural linked list.
- Duplicate of keys are maintained and all nodes are present at the leaf.
- Height is lesser than B tree for the same number of nodes.

Group - C

Ques 7 (a) Why circular queue is better than simple queue?

Ans - A circular queue is better than a simple queue because it has better memory utilization.
→ In a circular queue, the last element points to the first element, creating a circular link.
→ This allows you to insert an element in the first position if the last position is full and the first position is empty.
→ It can be used for both FIFO and LIFO struct

(c) Write a program to convert infix expression to its equivalent postfix expression using stack

Ans
```
#include <stdio.h>
struct stack {
    char data;
    int top;
};
struct stack *s;
```

```
void push (char c) {
  if (s->top == 100) {
    printf ("stack overflow! \n");
    exit (1);
  }
  s->top++;
  s->data [s->top] = c;
}
char pop() {
 if (s->top == -1) {
   printf ("stack underflow! \n");
   exit (1);
 }
 return s->data[s->top--];
}
int isoperand (char c) {
return (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'z');
}
int precedence (char c) {
 switch (c) {
   case '+';
   case '-';
     return 1;
   case '*';
   case '/';
     return 2;
   case '^';
     return 3;
}
```

```
return -1;
}
void convertInfixToPostfix(char *infix){
char *postfix = (char *)malloc(sizeof(char)*
                    strlen(infix));
int i,I;
i=I=0;
s = (struct stack*)malloc(sizeof(struct stack));
s->top=-1;
while(infix[i] != '\0'){
if(isoperand(infix[i])){
  postfix[I++] = infix[i++];
}else{
  while(s->top != -1 && precedence(infix[i])
<= precedence(s->data[s->top])
  postfix[I++] = pop();
}
push(infix[i++]);
}
}
while(s->top != -1){
  postfix[I++] = pop();
}
postfix[I] = '\0';
printf("The postfix expression is : %s\n",postfix);
}
int main(){
char infix[100];
```

```
printf("Enter the infix expression: ");
gets(infix);
convertInfixToPostfix(infix);
return 0;
}
```

Ques 8 - Write the advantages disadvantages and uses of a circular linked list.

Uses

→ Circular linked list can be used to implement queues, which are data structures that stores a collection of element in a first in (FIFO) order.
→ It is used to implement stacks
→ It is used to implement dequeues
→ It is used to calculate cyclic redundancy check (CRC) values.

Advantages

→ Insertion and deletion at the beginning or end of the list can be performed more efficiently than in singly linked lists.
→ The circular structure allows for continous traversal of the list without encountering a null value.
→ Circular linked list can be more memory efficient than arrays, as they do not require a separate pointer to keep track of the end of the list.

Disadvantages

→ The circular struct introduces additional complexity in terms implementation an maintenance compa to singly linked list.
→ Reverse traversa of a circular linke list can be more difficult than reverse traversa of a singly linked l
→ Random acce to elements in a circular linked lis is not possible, as there is no way to directly access the middle of list

Ques 9(a) What are the best worst and average case time complexity of binary search algorithm?

Ans- The time complexity of a binary search algorithm is O(log N). Best case → O(1), Avg case → O(log N) Worst case → O(log N).

(b) How linear search is differing from binary search?

Ans-

Linear Search	Binary Search
→ Commonly known as sequential search.	→ Commonly known as half-interval search.
→ Elements are searched in a sequential manner (one by one).	→ Elements are searched using the divide and conquer approach.
→ The elements in the array can be in random order.	→ Elements in the array need to be in sorted order.
→ Less Complex to implement	→ More Complex to implement
→ Linear Search is a slow process.	→ Binary search is comparatively faster.
→ Single and Multidimensional arrays can be used.	→ Only single dimensional array can be used.
→ Does not Efficient for larger arrays.	→ Efficient for larger arrays.

(c) What are the preconditions for performing binary search in an array?

Ans- The preconditions for performing binary search on an array are :—

- The array must be sorted in either ascending or descending order.
- The lower bound and upper bound of the array must be known.
- The comparison operator must be deterministic i.e., it must return the same result for the same input values every time.

(d) Write a C program for linear search in an array.

Ans-
```
#include <stdio.h>
int main() {
int array[] = {10, 20, 30, 40, 50};
int size = sizeof(array) / sizeof(array[0]);
int search_element = 30;
int index = -1;
for (int i = 0; i < size; i++) {
if (array[i] == search_element) {
index = i;
break;
}
}
if (index == -1) {
printf("The element %d was not found in the array.\n", search_element);
} else {
printf("The element %d was found at index %d.\n", search_element, index);
}
return 0;
}
```

Ques 10 (a) What is a stack ADT? Explain their operations.

Ans- A stack is an abstract data type (ADT) that stores a collection of elements. It follows a LIFO principle, where the most recently added element is deleted first.

A stack has two main operations:-

- Push: Adds an element to the collection.
- Pop: Removes the most recently added element that was not yet removed.

(b) Write an algorithm to delete an item from a circular array of queue: Also find the complexity of the algorithm.

Ans- Step 1: Check if the queue is empty. If it is, then return an error message.

Step 2: Calculate the new front index. The new front index is the old front index plus 1 modulo the size of the queue.

Step 3: Delete the item at the old front index.

Step 4: Update the front index.

Step 5: Return the deleted item.

The complexity of the algorithm is O(1).

Ques 11(a) Define algorithm.

Ans- An algorithm is a set of instructions that are followed in a specific order to perform a task.

(b) What are the measures of performance of an algorithm? Explain.

Ans- Time and space complexity are the two main measures for calculating algorithm.
(An algorithm must be analyzed to determine the resource usage of the algorithm. An algorithm's efficiency is referred to as the number of computational resources used by the algorithm.

Ques- Define asymptotic notation and explain each types.

Ans- Asymptotic notations are mathematical notations used to describe the running time of an algorith They are used to analyze and compare the runtimes of different algorithms

There are three types of asymptotic notations:

- Big-O: Used for upper bound values.
- Big-Omega: Used for lower bound values.
- Big-Theta (Θ): Used when lim $n \to \infty$ (f(x) g(x))=