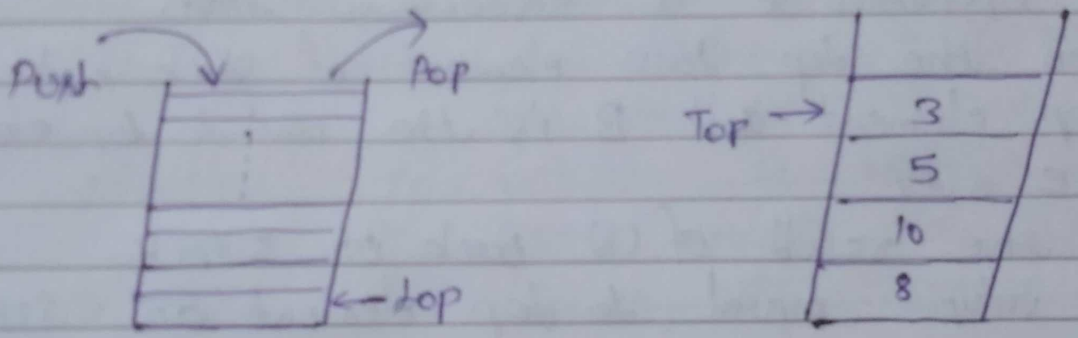
Tutorial Sheet -2

KCA-205 DATA STRUCTURES AND ANALYSIS OF ALGORITHM

Q.1:- Show memory representation, application and operations of stack with function in C.

A stack is one of the most commonly used data structure. A stack also called Last in First out (LIFO) system, is a linear list in which insertion and deletion can take place only at one end, called top.



There are two operation perform in stack

1) Push :- for insert element.
2) Pop :- for deletion of element.

Application of stack :-

1) Expression evaluation.
2) Expression conversion
3) syntax parsing
4) Parenthesis checking
5) string reversal
6) Function Call.

Q.2) Explain with suitable example

I) Postfix evaluation:-

1) Add a right parenthesis ")" to P. (This acts as a sentinel)
2) Scan P from left to right and repeat step 3 and 4 for each element of P until the sentinel ")" is encountered.
3) If an operand is encountered, put it on STACK.
4) If an operation ⊗ is encountered then
   a) Remove the top two element of stack, where A is the top element and B is the next to top element
   b) evaluate B ⊗ A
   c) Place the result of (b) back on STACK.
5) Set value equal to top element on STACK
6) END.

2) Infix to postfix conversion:-

1) Push "(" into STACK, and add ")" to end of Q.
2) Scan Q from left to right and repeat steps 3 to 6 for each element of Q until the STACK is empty.
3) If an operand is encountered, add it to P
4) If a Left parenthesis is encountered push it on to STACK.
5) If an operator ⊗ is encountered, then
   a) Repeatedly pop from STACK and add to P each operator (on top of STACK) which has the same precedence as or higher precedence than ⊗.

b) Add ⊗ to STACK.

6) If a right parenthesis is encountered, then.

a) Repeatedly pop from STACK and add to P each operator (on the top of STACK) until a parenthesis is encountered.

b) Remove the left parenthesis

7) END.

Q. 3) Explain sparse matrix with example. Backtracking.

A matrix can be defined with a 2D array. Any array with m column and n rows represent a m x n matrix. There may be a situation in which a matrix contains more number of zero values than non-zero. Such matrix is known as sparse matrix.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix} \rightarrow$$

Rows	Columns	values
5	6	6
0	4	9
1	1	8
2	0	4
2	3	2
3	5	5
4	2	2

Q. 4) List the areas of applications of sparse matrix.

Sparse matrices can be useful for computing large scale applications that dense matrices

can not handle. one such application involves partial differential equation by using the finite element method.
application of sparse matrices is in data structure and computer programming

Q.5):- what are the disadvantages of simple queue? How circular queue is better than simple queue

The main limitation of queues in a data structure is one of the basic operations of the deleting an element from it is cumbersome.
And has limited space
A circular queue is better than a linear/simple one because the number of elements the queue can store is equal to that of the size of the array.
In circular queue, there is no wastage of memory as it uses the unoccupied space and memory is used properly in a valuable and effective manner as compared to a linear queue.

Q.6) Differentiate between Queue and dequeue.

Deque	Queue
As the deque is a sequence container, so the insertion of elements takes place from both the end front and rear.	insertion of the elements take place from the one end from rear.

In deque, deletion of elements take place from both the ends front and rear.	In queue deletion of elements take place from the front only.
element can easily be accessed the deque using the iterators	It is not possible to access the elements of queue using the iterators.
Insertion and deletion of elements are more efficient as the resources are completely utilized in the case of the deque.	Insertion and deletion of elements are not much efficient in queue as compared to deque.

Q.7) Elaborate the concept of priority queue with suitable examples

A priority queue is an abstract data type that behaves similarly to the normal queue except that each element has some priority, the element with highest priority would come first in a priority queue. The priority of the elements in a priority queue will determine the order in which elements are removed from the priority queue

for example:- Suppose we have some values like 1, 3, 4, 8, 14, 22, inserted in a priority queue with an ordering imposed on the values is from least to the greatest. Therefore the 1 number would be having the highest priority while 22 will be having the lowest priority.

Q.8) Explain the following with example:-

(a) linear search:-

A linear search or sequential search is a method for finding an element within the list. It sequentially checks each element of the list until a match is found or whole list has been searched.

(b) binary search:-

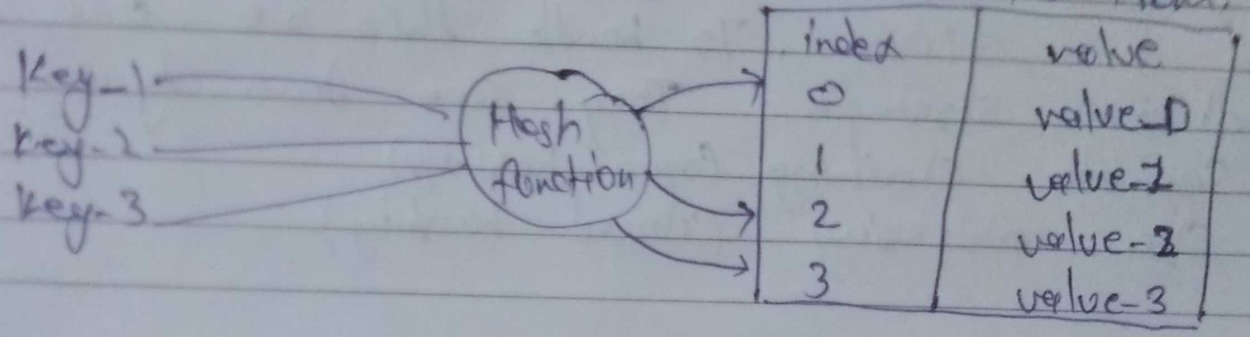
Binary search is an efficient algorithm for finding an item from a sorted list of items. It works by repeatedly dividing in half the portion of the list that could contain the item until you've narrowed down the possible locations to just one.

Q.9) :- what do you mean by hash table? explain with example.

Hash table is a data structure which stores data in an associative manner. In a hash table data is stored in an array format, where each data value has its own unique index value. Access of data becomes very fast if we know the index of the desired data.

~~Hashin~~ Thus it becomes a data structure in which insertion and search operations are very fast irrespective of the size of the data. Hash table uses an array of

as a storage medium and uses hash technique to generate an index where an element is to be inserted or is to be located from.



Q.10):- Describe the following with example:-

a) Hashing :- Hashing is a technique to convert a range of key values into a range of indexes of an array. We're going to use modulo operator to get a range of key values. Consider an example of hash table of size 20, and the following items are to be stored.

(1,20) (2,70) (42,80), (4,25), (12,44) (14,32) (17,11), (13,78) (13, 98)

S.No	key	Hash	Array index
1	1	1 % 20 = 1	1
2	2	2 % 20 = 2	2
3	42	42 % 20 = 2	2
4	4	4 % 20 = 4	4
5	12	12 % 20 = 12	12
6	14	14 % 20 = 14	14
7	17	17 % 20 = 17	17
8	13	13 % 20 = 13	13
9	37	37 % 20 = 17	17.

(b) Collision Resolution strategies:-

when two or more keys are given the same hash value, it is called a collision. To handle this we use collision resolution strategies. They are two types,

1) Separate chaining (open hashing)
2) Open addressing (closed hashing).

Short ANSWER TYPE QUESTION.

Answer No.1:- Push and pop.

Answer No.2:- when new data is to be inserted into the data structure but there is no available space. free storage list is empty this situation is called overflow.

Ans 3:- A pile of books, a stack of dinner plates, a box of pringles potato chips.

Ans 4:- A circular queue is the extended version of a regular queue where the last element is connected to the first element.

Ans 5:- Circular queue offers a quick and clean way to store FIFO data within a maximum size.

Ans 6:- An error condition that occures when an item is called from the stack but the stack is empty.

Ans 7:- The process of finding the required information from a collected of items stored as elements in the computer memory.

Ans 8):- when more than one value be hashed by particular hash function hash to the same slot, it is called collision.

Ans 9):- basic requirment of binary search is data is in the sorted form.

Ans 10):- There are two pointers used in dequeue.
LEFT and RIGHT.