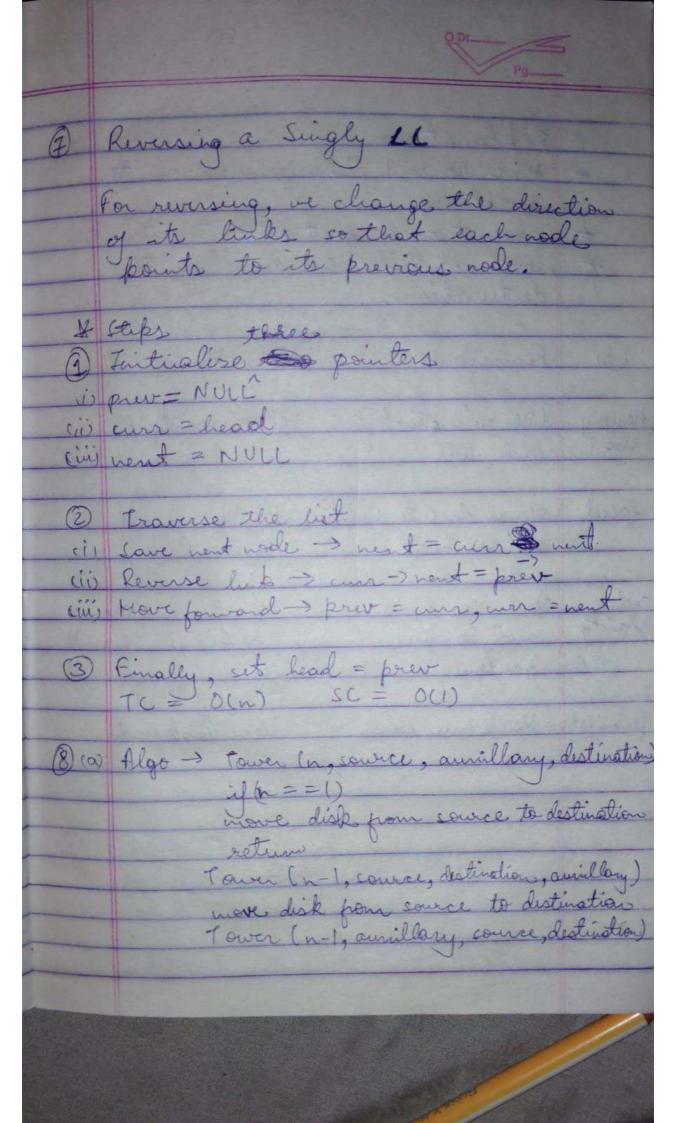
DSA MTE O Asymptotic Notation It describes how an algorithm's running time or space grows with time input sire non longe in is Big ? -> best case in Big @ -> average case flead Recursion (2) Tail Recursion Recussive call occurs learnine call Eg > ascending order descending order Address of A[in][ji] = B + W * (in x n+j (3) Linear Search -> Searches for an elevent Linearly -> one by one > Ocur Binary Searl -> Searlies for an elevery by dividing the array into smaller parts, checks the stements until dividing. TC > 0 (log)n)

that Insertion Sort Algorithm vii Start from the 2nd element [i=]] iii) compare it with elements on to one position to the right. (V) Place the element in it correct position Wo Move to the next element and refeat until the whole list is conted. Sparse diatrise has O elements. storing all elements wants menory, therefore in store only They are represented through



Enplanation * Hove top n-1 disks from source to more not largest disk to destinates more not disks from availlary to destination. Eteration (b) Remision Henry Uses call stack for Uses constant each call speed slower due to function Faster, no call calloverhead. code Essien for divide clarity and conquer problems. Sometimes more Termination Needs base condu. Uses look condu

(a) Algo merge Sort (an, l, r); y(l < r) merge sort (an I, mid) merge Sort (am, mid of, r) merge (an, l, mid, r) TC = O(n legn) SC = O(n) * Benefits over Rubble sont

i) there cont is faster

ii) Works efficiently on large datasets

iii) Itable and predictible performance D Time space trade off is the concept where sometimes more memory (space) can be used to make a process faster and vice -versa, for eg. -> Storing values Mutidimensional, aways are useful for refresenting naturin data, images int [71] matrin = { 11,23, 13,43}

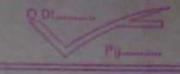
Aus-3 For a 1-D array an, element ancidis This formula is important as it allows Ans 4 int LS (int () ans, int value) { for (int i=0; i can longth; i+1) } returni) Adv -> Simple and works for any array. Ans is fluid sont picks a privat, positions the array so left are smaller and night are larger and quick conts each partition Ams & Direct Recursion -> function calls itself directly. Indirect Recursion > Function calls original . which then calls

void direct Rec (int n) & if (n70) direct Rec (n-1); vaid recA(intn) & ij(n>0) void rec B (int n) & if (n>0) sec Aln-D; 3 Insertion operation creates new node adjust privious / new pointers of neighbouring modes. Deletion operation removes node, adjust pointers to "skip" deleted node (8) (a) int fibonacci (int n) & y(n (=1) return n; int a = 0, 6 = 1;ii for (int i=2; i (=n; i++) (Recursion for Fibonacci can be replaced with above look, saving stack space.

9 store only non-zero elements as nodes containing now / column / id Renefits - source wemony compared to standard away for large, most no matrices. SET-3 1 Algorithm efficiency refers to how well an algorithm uses speces resources, especially time and space such as harjost it runs or how much memory it uses. It is measure using time complexity & space complex Dr. TC shows how running time increase with input size. In linear search O(n) as it may can every element (ii) SC describes how much memory is used (eg. an entra array in m 3 Inden = K*n*y+j*n+i (9 is Find the middle of the sorted array

If the middle is the target return its under. in If the target is smaller, repeat search in left half, if larger, then in right half. D Bubble sort refeateally compace adjacent elements and swap it out of order After each pas , the largest unorted element "bubbles" to the end. void bubble sort (int () and for (int i = 0; i k arr leigth -1, it) e for (int j=0: j (an light -1; jet) ? sig (an (j) > an (j+17)° int temp = an [j]; an (j) = an (j+1); ancitiz= top;

6 The factorial of Niln!) is int factorial (int n) & return + factorial (n-1); (8 (a) Discuss trade of 6/w recursion and iteration (memory) * Recursion uses more memory since each call adds to the call stack. * Iteration wes a fixed amount of memory, which is more efficient. (int tibonacci (int n) & return filonacci (n-1) + filonacci (n-2); 9x Herge Sort divides the array into holies sorts each, and merges the results * Used in enternal sorting (like in disk) large data sets or where stability is



SET-4

An algo is a finite sequence of steps to solve a problem. Efficiency is important because it determines the time and space complexity needed by algorithm, impacting performance especially for large datasets.

2 In now major order, elements of a row and stered in contiguous memory locations, in column-major order elements of a column are contiguous.

Diagram chould show 2D array labelled, to indicate access pottern.

Jor each scheme.

3 for an n-dimensional array sorted sorted in row-major order.

Address = Base + & (i current * product)

This formula calculates the linear wemony location for element.

De Justion cont can be more officient for partially sorted, data, while

selection sort does fewer steps but of comparisons.

Both have O(n2) time complexity in worst and average cares 5) Herge Sort is a divide- and-conquer algorithm. It divides the array into holves recursively sorts them, and merge Recursion can often be replaced with loops for efficiency. int factorial (int n) { for (inti=L; ikeh; itt) & return result;