BCS42 or DAA or Algorithm Notes by Ankit Kumar

Find the largest no. in the given 10 numbers

Find the sum of all 5 numbers

String concat

```c
#include <stdio.h>
int main()
{
   char first_string[20]; // declaration of char array variable
   char second_string[20]; // declaration of char array variable
   int i;  // integer variable declaration
   printf("Enter the first string");
   scanf("%s",first_string);
   printf("\nEnter the second string");
   scanf("%s",second_string);
   for(i=0;first_string[i]!='\0';i++);
   for(int j=0;second_string[j]!='\0';j++)
   {
      first_string[i]=second_string[j];
      i++;
   }
   first_string[i]='\0';
  printf("After concatenation, the string would look like: %s", first_string);
return 0;
```

Numer is odd or even

```c
#include <stdio.h>
int main() {
int number;
printf("Enter a number: ");
scanf("%d", &number);
```

```
    if (number % 2 == 0) {
printf("%d is an even number.\n", number);
    } else {
printf("%d is an odd number.\n", number);
    }
    return 0;
}
```

ALGORITHM

set of instruction hota hai
jo computer ki particular problem ko solve karta h

Adv

Easy to understand
Har problem ko step by step solve karte h

***ALGORITHM KA CHARATER**

1. Input:

User ke diye hue data per kam karta hai aur problem ko solve karta hai

2. Output:

Hamara algorithm problem ka solution produce karta hai

3. Finiteness:

Algorithm ko kuchh steps ya kam time mein complete hona chahie

4. Generality:

Algorithm same problem ke various interface per kam karna chahie

5. Conciseness:

Algorithm simple or easy to understand hona chahiye

***MATHMATICAL PROBLEMS**

basic calculations like addition, subtraction, multiplication, and division.

Search problems:

Ek item ko list mein se find karna
aur do cities ke bich mein se ek shortest route find karna

Sorting problems:

Numbers ki list ko assending aur descending order mein arrange karna chahie

Optimization problems:

Problem ka best solution find karna

Recurrence relation

jo ek series (sequence) ke terms batata hai
ek mathematical equation hota hai
jo sequence me har agle number ko pehle ke numbers se jod kar batata hai.
$F(n)=F(n-1)+F(n-2)$

Notation

ye batata hai koi kaam kitni jaldi or efficient hota h
ye performance ko describe karta hai
notation ek chota code hota hai

Asymptotic Notetimes

ye batata h Ek algorithm dusre algorithm se kitna bhetar h
ye mathematics ka tool hota h
ye complexity ko represent karta h,
Big O, Big Omega, Big Theta

Big O

Algorithm ki efficiency ko input size ke sath compare karta hai
asymptotic upper bound ko describe karta h
worst case me use kiya jata h

$f(n) <= Cg(n)$ for all $n >= n0$

Big Omega

asymptotic lower bound ko describe karta h
best case me use kiya jata h

$f(n) >= Cg(n)$ for all $n >= n0$

Big Theta

average case me use Kiya jata h

$f(n) <= C1g(n)$ for $n >= n0$

$f(n) >= C2g(n)$ for $n >= n0$

Merging both the equation, we get :

$C2g(n) <= f(n) <= C1g(n)$ for $n >= n0$

Worse case

Ek function maximum no. Of steps perform karta h

Best case

Ek function minimum no. Of steps perform karte h

Average case

Ek function average steps perform karte h

Complexity

ek measurement hota hai
koi bhi operation karne me kitna time or space lagta hai

jisse programming ko efficient banate hai
Time complexity
Hamara program kitna time leta hai
ek specific task ko complete karne me
Space complexity
Program ko kitni extra memory ki jaroorat Hoti hai

Divide
Ek problem ko chote sub problem me divide karte h
Conquer
Sub problem ko solve karte h recursion method se
Combine
Sub problem k solutions ko ek final solution me combine karta h

Knapsack
ek bag mein saman bharte hi ki bag ka total value sabse zyada ho aur weight limit ke andar k andar rahe

Bubble sort
array me adjacent elements ko compare kiya jata h
aur agar wo galat order mein hain toh unhe swap kiya jata h
isme iteration steps ka use Kiya jata h
Bubble sort ek sorting algorithm hai

Quick sort
array ko chote parts me divide karte hai
phir har part ko sort karte hai
phir ihne merge karke sorted array banalete h
divide and conquer k principal par kam karta h
Isme pivot, p aur q let kiya jata h
ek sorting algorithm hai

Selection Sort
unsorted list me repeatedly minimum or maximum element ko select karta hai
Aur sirf ek time me next element se compare karke beginning or end me move karta h Ek sorting algorithm hai

Heap sort
Ye selection sort ki tarah similar hota hai
minimum element ko find karta hai
aur shuruaat mein rakhta hai
isme min or max operation hote h jo bhot fast hote h jo element ko add or remove karne k liye use Kiya jata h

insertion sort
array k andar har ek element ko sorted sequence me rakhte hai
insertion sort ek sorting algo. Hai
jese playing cards ko ek ek karke apni sahi position par rakhte hai

merge sort
isme data ko do parts me divide karte hai
phir each part ko recursively divide karte h
phir har individual part ko sort karte h
aur last me sorted parts ko merge karkar sorted single array banate h
iska principal divide and conquer algorithm hai

Genetic Algorithm
ek search aur optimization technique hai
jo natural selection aur genetics ke principles ka use karke optimization aur search problems ko solve karta ha

***Recursive algorithm**
Ek particular problem ko chhote sub problem mein solve karta hai
khud ki copy ko call karke karta hai

```
def factorial(n):
    if n == 0 or n == 1:
        return 1
   n * factorial of (n-1)
    else:
        return n * factorial(n - 1)
print(factorial(5))
```

```
factorial(5)
  |
5 * factorial(4)
  |
4 * factorial(3)
  |
3 * factorial(2)
  |
2 * factorial(1)
  |
1
```

***Strassen's Algorithm**

Strassen's algorithm बडी matrices को गुणा करने का एक तेज तरीका है
जो standard तरीके से ज़्यादा efficient है.
do matrix ko guna karne ke liye "divide and conquer" method ka istemal karta h
Jisme p,q,r,s,t, u,v formules hote h

Karatsuba algorithm
bade numbers ko small numbers me tezi se multiply karta h
Multiplication karne ka fast tarika h
ye divide and conquer approach algorthm hai.
S3-s2-s1

Bellman-Ford algorithm
ek starting point se graph ke sabhi points tak ka shortest path find karta hai, chahe kuch weights negative bhi ho.

***Greedy algorithm**
हर स्टेप पर सबसे best aur immediate solution select karta h
ये हर बार सही जवाब nhi deta
लेकिन कई समस्याओं का जल्दी solve karta h
Greedy about profit
Greedy about weight
Greedy about both
Greedy algorithm example
Prism algorithm
Kruskal algorithm
Travelling salesman algorithm

***Searching algorithm**
Ek dataset me Ek aur do se jyada element ko sequence me search karte h
Types of Searching algorithm
Linear search algorithm
Binary search algorithm

Linear search
Array me har ek element ko sequence me check karte hai aur jab element mil jata hai to use locate kar dete hai ya index return karedete h

Binary search
sorted array me middle me se element ko dhundna start karte hai
agar element middle se chota hai to uski first half me dhundte hai nhi 2nd half me
binary search divide or conquer technique hai

***Dynamic programming**
ek problem solving approach h
Jisme big problems ko chote chote sub problems me divide karte h
phir inko solve karte h phir store karte h phir baad me reuse karte h

BFS
Breadth first search
nodes ko level by level explore karte hain,
shuruwat source node se hoti hai.
Isme queue ka istemal hota hai

DFS
Depth first search
Ek node se shuru karte hain
aur uske adjacent nodes ko puri depth tak explore karte hain
phir doosre adjacent nodes ki taraf badhte hain
Isme recursion or stack ka use hota hai

****Minimum spanning tree***
nodes ko jodta h
par sabse kam cost ka total karta h
jis mein sabhi vertics ka total cost
sabhi spanning tree ke total cost se kam hota hai.

Dijkstras Algorithm
Map में दो place के beech shortest path find karta h
Applications:
Game development, Circuit design, Robotics, Transportation, Find location in map

***Prims algorithm**
Mein kisi bhi vertex se shuru karte h
Vetics ko badhate hue mst banate h
prism algo. Is a minimum spanning tree
Graph mein Minimum edges ka set find karta h jo sabhi vertics se connected hota h
Applications electrical wiring, traffic management, fraud detection, sales man

***Kruskal's Algorithm**
Sabse chote edges se shuru karte h
jo edges k weight ko ascending order mein select karke
bina cycle banaye graph ko connect karta.hai
Kruskal's Algorithm ek graph ka minimum spanning tree h
Applications - network design, circuit design, robotics, lan - local area network

Tree traversal
jab tree ko explor karte hai tab tree ko specific order me visit karte hai
ise 2 category me classified kiya jata h
har node ko sirf ek baar visit karte hai
Types

Inorder LNR
Phele left subtree ko visit karte h
Phir root node ko
Phir right subtree ko dekhte h
Preorder NLR
Postorder LRN

***Skip list**
ye linked list ki tarah kaam karta h
sorted list h, jo linked list se better h
or jaldi search performance karta h
एक डाटा structure h jo element ko linked list se jaldi dhundte h

hash function
jo kisi bhi data ko short or fixed / unique code me badal deta h
hash algorithm, special function h

***Encryption Algorithm.**
jo data ko unreadable bana deta h ya gupt kar deta h
taki use sirf authorised log hi access kar saken ya pad sake

****cryptographic algorithm***
data ko unreadable code me badal/ scramble kar deta h secaurity k saath
bina authorisation k convert karta h,

****Radix sort algorithm***
ye numbers ko unke individual digits se sort karta h
Radix sort is a digit-by-digit sorting technique hai
numbers ko ek-ek digit ke hisaab se sort kiya jata hai
shuruwaat sabse chote waale digit se karte h

***Direct recursive algorithm.**
ek function sidhe khud ko call karta hai.
khud ko dubara call karta hai
jab tak ki koi base case na mil jaaye.

***inDirect recursive algorithm.**
jismein ek function khud ko sidhe call nahin karta hai
dusre function se khud ko call karta hai

***Control abstraction**
program control ki complex chije ko chupate h aur asan tarika provide karte h

***Feasible solution**
solution ka problem ki condition ko pura karta h

****optimal solution.***
kisi problem का सबसे *bheter solution* होता है
जो सभी *condition* पूरा *karta h aur* सबसे अच्छा *outcome* देता है।
kam resource me best outcome deta h

***backtracking**
isme alag alag option try kiya jata h
aur agar valid solutoin find nhi ho pata to backtrack kiya jata h
dfs ka rule follow karta h,
Iske solution ko tree ki form me represent karte h
Ex chess

***implicite**
jo clearly bataya nhi jata lekin algorithm ke kam karne ke tarike se samajh jata h

***explicite**
jo sidha bataya jata aur algoridam ke kam karne ke tarike se bhi samajha jata h

***dynamic tree.**
tree k structure ko change aur modify kiya jata h

***Horner’s rule**
Horner's rule polynomials equation par use kiya jata h
no. Of multiplications ko reduce karta h

***static tree**
tree k strucure ko change nhi kiya ja sakta h
but iske andar information ko change kiya ja sakta h
jiske node aur connection fixed hota h
element ko remove nahi kar sakte h

insertion sort
Time Complexity Worst-case and average-case: $O(n^2)$, Best-case: $O(n)$

Space Complexity O(1) (constant

selection sort
Space Complexity: O(1)
Time Complexity: O(n^2)

Binary search algorithm
time complexity O(log n)

quick sort worst case
Quicksort का worst case तब होता है जब हर बार pivot चुनने में सबसे छोटा या सबसे बड़ा element चुन लिया जाए