

| Date : Page No. : |
|--|
| |
| Write down classification of Data Structure in detail. |
| |
| and organize data. |
| o and a adding a |
| it can be accessed and uploade updated efficiently. |
| it can be accessed and uplands updated and |
| - O |
| Data Structure is uny of oxpanising all its it |
| -> Data Structure is usy of oxganising all data item and relationship to each other. |
| AVANTA MARKATANA |
| Types of data structure: |
| MOR CATALOGY IN THE |
| Carlotte And Andrews |
| Data Structure |
| And the second s |
| Poimitive data-structure |
| Integer Won-Primitive data structura |
| Character |
| Float Linear D.S. Non-linear D.S. |
| Boolean Non-Lineax O.S. |
| |
| Static Dynamic - Graph |
| Axxau |
| STACK |
| Linked List |
| Queue |
| |



N State of S

SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

Date : Page No. : (1) Primitive Data Structure: There are basic structure and are directly. operated by machine instructions. Fox example: Integer, float, Character (\mathfrak{I}) Non-Primitive Data Structure: They are desive from the Primitive data Structure. It is a collection of some data type ox different data type Primitive data Structure. - These data structures can't be manipulated or operated directly by machine-level instructions For example: - Array, trees & Graph. > Lineax Data structure :-The assangement of data in the sequential mannex is Known as Linear data structure.
The data structure used for this Purpose are Arrays, Linked List, Stacks and Queves. - In this data structure one element is commented to only one another element in a linear form.



| 1 | Date : Page No. :3 |
|-------|--|
| | - Data Staucture can also be classified as Static ox Dynamic data Staucture. |
| | 7 Non-Linear Data Structure: |
| | - When one element is connected to the 'n' number of elements is known as non-linear D.S. |
| | - In this case, elements are arrage in random method. |
| | For example: Trees and Graph. |
| Que:2 | |
| | What is Algorithm? Explain its types with example. |
| Las - | -An algorithm is a Process ox set of rules required to |
| | detinition of |
| | finite Set of instructions which are being carried in a Specific order to perform the |
| | 110 5000 |
| | - It is not the the complete Program. ox code: it is net a solution (logic) of a Problem, which can be refresented either as an informal description |
| | a Flowchart or Pseudocode. |
| | |
| | |



| , v | Date : Page No. : |
|--|--|
| * | Types of Algorithm: |
| | |
| | |
| (1) | Sequence Algorithm: |
| | - Justinia |
| l la | - This algorithm Dorfon Guerra |
| | - This algo withm Pexform Successively one by one without skipping any steps. |
| | The steps. |
| 71 V T 1 - | - No soloction Donald |
| | - No selection Procedule or condition branching exits in sequence algorithm. |
| | The sequence algorithm. |
| | Example: |
| | TO STORY OF THE ST |
| | Madding two numbers |
| | Leo nombeos |
| 4,14, 14 | Step1: Start |
| | Step 2: read a, b |
| , , | SteP3: -Sum = a76 |
| ÿ | Step4: Write Sum |
| • | Step 5: Stop. |
| | |
| (2) | Selection Algorithm: |
| , | |
| | - The Sequence type of algorithm are not sufficient |
| , | to solve the Problem which in values decisions and |
| | conditions |
| , | |
| | - In order to solve the Problem which involve |
| | - In order to solve the Problem which involve decision making or option selection we go for selection type of algorithm. |
| | Selection tyre of algorithm. |
| | |

| 3 | The second |
|---|------------|
| | (4) |
| | |

| TITL THE YOUR | EDUCATION TO INNOVATION Date: |
|---------------------------------------|--|
| | Example: |
| | - Cumpie. |
| | (0, 0) |
| | Algorithm Smaller (A, B) A, B: Values to be checked. |
| | H, B: Values to be checked. |
| | TO(h) = 0 |
| | Step1:- If (A TB) |
| | Return A |
| | |
| FF. | Step2:- Else |
| | Return B |
| (-2 | The state of the s |
| (3) | Desation algorithm: |
| | |
| | - Iteration type of algorithm are used in solving Problem which involves repeation of statement. |
| · · · · · · · · · · · · · · · · · · · | which involves repeation of statement. |
| | |
| | - In this type of algorithms, a Particulars number |
| | of statements are repeated 'n' no. of times. |
| (, | |
| | Example: |
| | |
| | Algorithm Print-n (n) |
| | (1) $T_{1,2}$ |
| | $\begin{array}{cccccccccccccccccccccccccccccccccccc$ |
| | (2) Repeat up to step until I <= N |
| | (3) Print T (4) T++ |
| | |
| | (5) Return |
| | |

| | DAK IINIVE |
|----------------------|--|
| | EDUCATION TO INNOVA- |
| Que | EDUCATION TO INNOVATION |
| Sue:-3 | Pate |
| | What is Time C. al |
| Ans | 1 0/00 () |
| | and time complete |
| | amount of time of an ale ill |
| | The time Complexity of an algorithm is the number of steps The time complexity of an algorithm is the number of steps The time Complexity of an algorithm is the number of steps The time Complexity of an algorithm is the number of steps The time complexity of an algorithm is the number of steps The time complexity of an algorithm is the number of steps The time complexity of an algorithm is the number of steps |
| | |
| | 106 7: |
| | denoted by all of an always |
| | The time complexity of an algorithm is denoted by the big o notation. |
| | Here I |
| | to reproce till on is the asymptotic |
| - | to represent the time complexity. |
| 11 (11) | example. |
| | GOLDEN GOLDEN |
| A | THE A COLUMN TO SEE A SECOND TO SECOND THE PARTY OF THE P |
| | #include <stdio.h></stdio.h> |
| 7 | V. T. D. C. V. T. |
| | Void Point Numbers (intn) |
| | |
| 15 | for (int i =1; i <= n; i++) |
| 7: | |
| | Point f. ("% d", i); |
| | 7 |
| | 9 |
| | int main () |
| | 9 |
| | int num 55; |
| 15 | Print Numbers (num); |
| | return O; |
| | & |
| Decree of the second | |
| | |



| THE TEN PENNY | EDUCATION TO INNOVATION Date: |
|---------------|--|
| Que: - 4 | |
| Ans. | - The amount of space Occupied by an algorithm Known as space complexity. |
| | - An algorithm is said to be efficient if its occupied less space and require the minimum amount of time to complete its execution. |
| | Example: |
| | #include Tstdio.h> |
| | Void Create Array (intn) |
| | int* axx = (int*) malloc (n* Size of (int)); |
| | for (int-i=0; i <n; i++)<="" th=""></n;> |
| K) | Account of the Control of the Contro |
| | ar [i] = i + 1; |
| | \mathcal{L} |
| | free (arr); |
| | int main () (|
| | int num = 5 ; |
| | Create Array (num); |
| | retum O: |
| |) |
| | \mathcal{G} . |
| | |



| जार पार्थ प्रतास | EDUCATION TO INNOVATION Date: |
|------------------|--|
| Q ve: 5 | Wite down Asymptotic Notation in detail. |
| ins | The Commonly used asymptotic notation used for calculating the sex running time Complexity of an algorithm is given below: 1 Big oh notation (0) 2 Omega Notation (0) 3 Theta Notation (0):- (11054 0056) Big oh Notation is an asymptotic that measures the Performance of an algorithm by Simply Providing the order of growth of the function (g(n)) |
| | n _o n |
| ·5 | |

| EDUCATION TO INNOVATION Date: | |
|--|--|
| $C_{2}(n) = term$ | |
| +(n) = Tonckion | _ |
| n = InPut Size C > 0, No:>1 C = (Positive num) | _ |
| C = (POSITIVE NUM) | |
| F(n) = O(g(n)) | ~ |
| | |
| → This measures the Performance of an algorithm by Simply & Providing the order of growth of the | The state of the s |
| function. This notation Provides an upperbound on a function which ensure that functions never grows faster | |
| than the upper bound I worst case. | - |
| Example:- If f(n) and g(n) are two functions defined | |
| for positive integers. | |
| then. f(n) = 0g(n) as f(n) is big oh of g(n) ox f(n) 15 on order of g(n) if there exists cont constants such that | _ |
| 15 on order of g(n) it there exists tom conseque | _ |
| C and no Such that. (and no Such that. (fin) (cg(n) fox all n > no | |
| | |
| | _ |
| | _ |

SEASON OF THE PARTY OF THE PART

SILVER OAK UNIVERSITY

| | Date: |
|--------|--|
| 64 | |
| 31 | Omega notation: (N) (Best case) |
| 14.5 | |
| | F(n) |
| 111 | |
| | -(g(n)) |
| | |
| à C | 11 11 18 41 41 41 14 14 14 14 14 14 14 14 14 14 |
| | No h |
| | A STATE OF THE PARTY OF THE PAR |
| | $f(n) \geq cg(n)$ |
| ~! h ! | $m \gamma_{mo}$ |
| | C70, Mo7, 1 |
| | TO CALLED IS 15 |
| | $F(n) = \mathcal{R}(g(n))$ |
| 4 | > It basically described best case scenarios |
| | -> It basically described best contation |
| , | 01/13 |
| 8,3 | + It is formal way to represent lower limit bound |
| . , | + o an algorithm running time. |
| | 1 C lima on algorithm |
| _ | -> It measures the best amount of time an algorithm can take to complete or best case time |
| | can take to complete |
| | · Complexity. |
| _ | |
| | |
| | |
| | |
| ı | |

| | Date : |
|---|--|
| | |
| 1 | Ecample: |
| 1 | |
| _ | If f(n) and g(n) are two functions defined for Positive |
| | integexs. |
| | |
| | then, f(n) = R g(n) as f(n) is omega of g(n) ox f(n) on the oxdex of g(n) if there exists constants c and |
| | the order of g(n) if there exists constants c and |
| | on airh that |
| _ | f(n) >= Cg(n) fox all n>no and c>o |
| _ | |
| 3 | Theta notation (0): (Average Case) |
| | |
| | (2gh) |
| | The state of the s |
| | C. gim) |
| | |
| | |
| I | |
| ı | |
| | $C_{1}g(h) = f(h) = C_{2}g(h)$ |
| ı | n y no |
| | C70, no 7, 1 |
| | F(m) = 0 (g(m)) |
| ۱ | The miles Average as a notation |
| | - The theta notation mainly describes Avexage ase notation |
| | - I't re Present realistic time complexity of an algorithm |
| _ | Big theta is mainly use where, the value of worst case and best case is came. |
| - | WORST CASE CARD DEST CONTRACTOR |