

DSA Introduction Part 2

TCS Exam Example

There is a JAR full of candies for sale at a mall counter. JAR has the capacity N, that is JAR can contain maximum N candies when JAR is full. At any point of time, JAR can have M number of Candies where $M \leq N$. Candies are served to the customers. JAR is never remain empty as when last k candies are left. JAR is refilled with new candies in such a way that JAR get full. Write a code to implement above scenario. Display JAR at counter with available number of candies. Input should be the number of candies one customer can order at point of time. Update the JAR after each purchase and display JAR at Counter.

Output should give number of Candies sold and updated number of Candies in JAR.

If Input is more than candies in JAR, return: "INVALID INPUT"

Given,

$N=10$, where N is NUMBER OF CANDIES AVAILABLE

$K \leq 5$, where k is number of minimum candies that must be inside JAR ever.

Example 1: ($N = 10$, $k \leq 5$)

Input Value

3

Output Value

NUMBER OF CANDIES SOLD : 3

NUMBER OF CANDIES AVAILABLE : 7

Example : ($N=10$, $k \leq 5$)

Input Value

0

Output Value

INVALID INPUT

NUMBER OF CANDIES LEFT : 10

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 29 of 49

Algorithmic Solution

Step1: START

2: DECLARE AND INITIALIZE VARIABLES... $N=10$ AND $K \leq 5$

3. ACCEPT INPUT FROM THE USER..

4. CHECK CONDITION..

5. IF TRUE: DISPLAY THE POSITIVE / ACCEPTED RESULTS

6. IF FALSE: DISPLAY THE NEGATIVE/ REJECTED RESULTS.."INVALID INPUT."

7. STOP

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

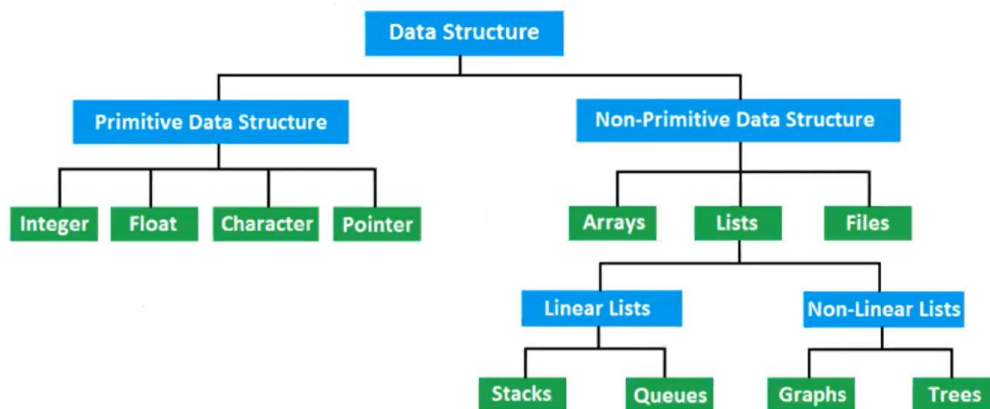
Need for Data Structures

- ❑ Data Structures organize data = efficient programs.
- ❑ The choice of data structure??
- ❑ A data structure requires certain amount of:
“Time , Space, Programming efforts.”
- ❑ Data Structures deals with the study of how data is organized in the memory, how efficiently the data can be retrieved and manipulated, and possibly the different data items are logically related.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 32 of 49

Classification of Data Structures



This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 33 of 49

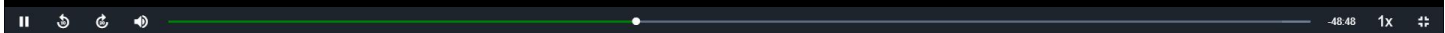
PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

Classification of Data Structures

- ❑ **Dynamic Data Structures:** grow and shrink during execution
- ❑ **Linked Lists:** insertions and removal made anywhere
- ❑ **Stacks:** insertions and removal made only at top of the stack
- ❑ **Queues:** insertion made at the back and removal made from the front
- ❑ **Binary Trees:** high speed searching and sorting of data
- ❑ **Graphs:** non-linear data structure consisting nodes and edges

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 34 of 49



PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

TCS Exam Example

Problem Statement: Checking if a given year is leap year or not

Explanation:
To check whether a year is leap or not

Step 1:

- We first divide the year by 4.
- If it is not divisible by 4 then it is not a leap year.
- If it is divisible by 4 leaving remainder 0

Step 2:

- We divide the year by 100
- If it is not divisible by 100 then it is a leap year.
- If it is divisible by 100 leaving remainder 0

Step 3:

- We divide the year by 400
- If it is not divisible by 400 then it is a leap year.
- If it is divisible by 400 leaving remainder 0

Then it is a leap year

leap year
a year, occurring once every four years, which has 366 days including 29 February as an intercalary day.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 35 of 49



Asymptotic Notations

- ❑ The efficiency of an algorithm is dependent on amount of time, storage and other resources. The efficiency is measured with the help of asymptotic notations.
- ❑ Asymptotic notations are mathematical notations used to describe the running time of an algorithm.
- ❑ Example: Bubble Sort:
 1. If input array is already sorted: best case
 2. If array is in reverse condition: worst case
 3. Array is neither sorted nor in reverse condition: average case

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Asymptotic Notations

There are mainly three asymptotic notations:

- **Big-O Notation (O -notation)** -----→ represents upper bound of running time of algo.
(worst case complexity)
- **Omega Notation (Ω -notation)** -----→ represents lower bound of the running time of algo.
(best case complexity)
- **Theta Notation (Θ -notation)** -----→ represents upper & lower bound of the running time of algo.
(average case complexity)

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video