

CHAPTER – 3
Data Structures and operations.(Expected Marks = 6)

(ഈ നോട്ട്സിൽ വിവരിച്ചിട്ടുള്ളതും ഉൾപ്പെടുത്തിയിട്ടുണ്ട്. പാഠഭാഗങ്ങൾക്കൊപ്പം വിവരിച്ചിട്ടുള്ള സൂത്രീയങ്ങളും കാണാവുന്നതാണ്. മൊബൈൽ ഫോണിലാണ് നിങ്ങൾ ഈ നോട്ട്സ് വായിക്കുന്നതെങ്കിൽ QR കോഡിൽ ക്ലിക്ക് ചെയ്യുക. Printed Note ആണെങ്കിൽ QR കോഡ് സ്കാൻ ചെയ്യുക.)

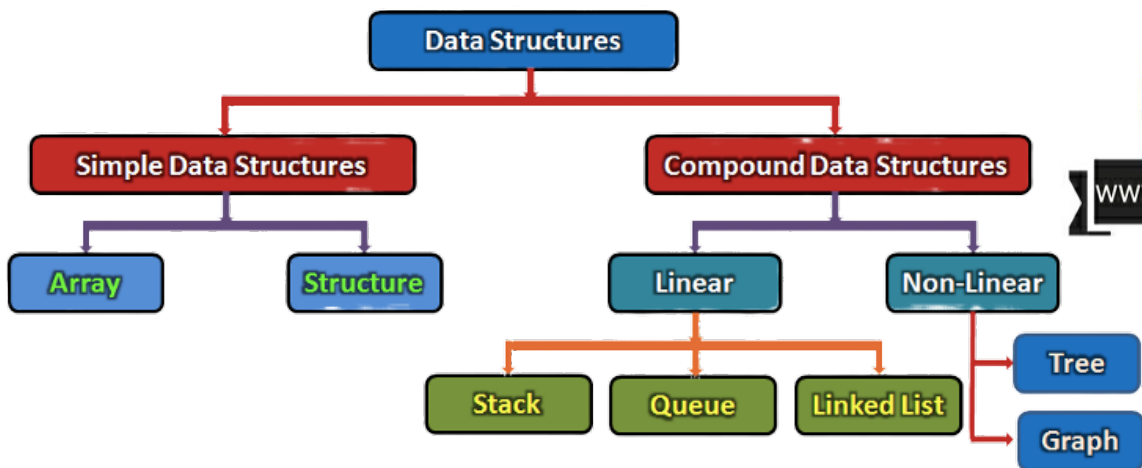
DATA STRUCTURE

- It is the way of organising data in the memory.
 (മെമ്മറിയിൽ ഡാറ്റ ക്രമീകരിക്കുവാനുള്ള മാർഗമാണിത്.)



Click or Scan here to see the Video

Classification of Data Structures



Operations on Data Structures

- 1. Traversal** - It is the process of visiting elements in a Data Structure
 (ഒരു ഡാറ്റാ സ്ട്രക്ചറിലെ ഓരോ അംഗത്തെയും സന്ദർശിക്കുന്ന പ്രക്രിയയാണിത്.)
- 2. Searching** - It is the process of finding the location of an element.
 (ഡാറ്റ സ്ട്രക്ചറിലെ ഒരു അംഗത്തിന്റെ സ്ഥാനം കണ്ടെത്തുന്ന പ്രക്രിയയാണിത്.)
- 3. Inserting** - It is the process adding a new data.
 (ഡാറ്റ സ്ട്രക്ചറിലേക്ക് ഒരു പുതിയ അംഗത്തെ ചേർക്കുന്ന പ്രക്രിയയാണ്.)
- 4. Deletion** - It is the process removing an element
 (ഡാറ്റ സ്ട്രക്ചറിലെ ഒരു അംഗത്തെ നീക്കം ചെയ്യുന്ന പ്രക്രിയയാണിത്.)
- 5. Sorting** - It is the process arranging elements in a Data Structure.
 (ഡാറ്റ സ്ട്രക്ചറിലെ അംഗങ്ങളെ ക്രമീകരിക്കുന്ന പ്രക്രിയയാണിത്.)
- 6. Merging** - It is the process of combining elements of two Data Structures.
 (രണ്ട് ഡാറ്റാ സ്ട്രക്ചറുകളിലെ അംഗങ്ങളെ സംയോജിപ്പിക്കുന്ന പ്രക്രിയയാണിത്.)





Click or Scan here to see the Video

STACK

- The Data Structure that follows **LIFO** principle is known as **STACK**.
LIFO തത്വം പിന്തുടരുന്ന ഡാറ്റാ സ്ട്രക്ചറിനെ STACK എന്നുവിളിക്കുന്നു.
- **LIFO** means **Last In First Out**
LIFO എന്നാൽ അവസാനമായി ADD ചെയ്തത് മാത്രമേ ആദ്യം DELETE ചെയ്യാനാകൂ എന്നതാണ്.
- It is a list of items in which all insertions and deletions are made at one end, known as **STACK TOP**.
(STACK TOP എന്ന് പറയുന്ന ഒരറ്റത്തുനിന്നു മാത്രമേ ഇനങ്ങളെ കൂട്ടിച്ചേർക്കാനും നീക്കം ചെയ്യാനും സാധിക്കൂ.)
- The element added at last will be the first to be removed from the **STACK**.
(സ്റ്റാക്കിൽ അവസാനമായി കൂട്ടിച്ചേർത്ത അംഗത്തെ മാത്രമേ ആദ്യം ഒഴിവാക്കാനാകൂ.)

Operations on STACK

PUSH

- Inserting an element into a STACK.
(സ്റ്റാക്കിലേക്കു ഒരു പുതിയ അംഗത്തെ ഉൾപ്പെടുത്തുന്ന പ്രക്രിയ.)

POP

- Removing an element from a STACK
(സ്റ്റാക്കിൽ നിന്നും ഒരംഗത്തെ ഒഴിവാക്കുന്ന പ്രക്രിയ.)

Algorithm of PUSH operation.

(അൽഗോരിതം പഠിക്കുന്നതിനു മുൻപ് വീഡിയോ കാണുക. ഗ്രാഫിക്സ് ന്റെ സഹായത്തോടെ അൽഗോരിതത്തിന്റെ പ്രവർത്തനം വിശദീകരിച്ചിട്ടുണ്ട്.)

Assume that STACK[N] is an array of stack with size N and TOS denotes the top position of the stack. Let VAL contains the data to be added into the stack. (സ്റ്റാക്കിൽ ആകെ എത്ര എലമെന്റുകളെ സൂക്ഷിക്കാൻ സാധിക്കും, എന്നതിന്റെ എണ്ണമാണ് **N**. സ്റ്റാക്കിന്റെ മുകൾ ഭാഗത്തെ സൂചിപ്പിക്കുന്ന വേരിയബിൾ ആണ് **TOS**. സ്റ്റാക്കിലേക്ക് add ചെയ്യാൻ പോകുന്ന value സൂക്ഷിച്ചിരിക്കുന്ന വേരിയബിൾ ആണ് **VAL**.)

- Start
- 1: If (TOS < N-1) Then
 - 2: TOS = TOS + 1
 - 3: STACK[TOS] = VAL
 - 4: Else
 - 5: Print "Stack Overflow "
 - 6: End of If
- Stop



Click or Scan here to see the Video



Algorithm of POP operation.

(അൽഗോരിതം പഠിക്കുന്നതിനു മുൻപ് വിഡിയോ കാണുക. ഗ്രാഫിക്സ് ന്റെ സഹായത്തോടെ അൽഗോരിതത്തിന്റെ പ്രവർത്തനം വിശദീകരിച്ചിട്ടുണ്ട്.)

Assume that STACK[N] is an array of stack with size N and TOS denotes the top position of the stack. Let VAL contains the data to be added into the stack. (സ്റ്റാക്കിൽ ആകെ എത്ര എലമെന്റുകളെ സൂക്ഷിക്കാൻ സാധിക്കും, എന്നതിന്റെ എണ്ണമാണ് N. സ്റ്റാക്കിന്റെ മുകൾ ഭാഗത്തെ സൂചിപ്പിക്കുന്ന വേരിയബിൾ ആണ് TOS. സ്റ്റാക്കിലേക്ക് add ചെയ്യാൻ പോകുന്ന value സൂക്ഷിച്ചിരിക്കുന്ന വേരിയബിൾ ആണ് VAL.)

Start

- 1: If (TOS > -1) Then
 - 2: VAL = STACK[TOS]
 - 3: TOS = TOS - 1
 - 4: Else
 - 5: Print "Stack Underflow "
 - 3: End of If
- Stop



Click or Scan here to see the Video

QUEUE (ക്യൂ)

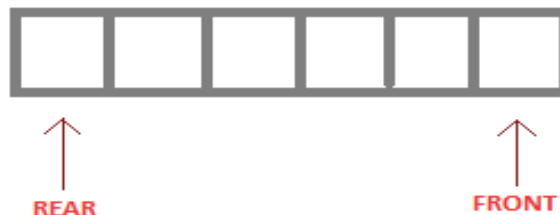
Click or Scan here to see the Video



- A Queue is a linear data structure in which insertion takes place at one end(**Rear end**) and deletion takes place at other end(**Front end**)

ക്യൂ എന്ന ഡാറ്റാ സ്ട്രക്ചറിൽ, എലമെന്റുകളെ add ചെയ്യുന്നത് പിൻ ഭാഗത്തു നിന്നും (REAR END), ഡിലീറ്റ് ചെയ്യുന്നത് മുൻ ഭാഗത്തുനിന്നും (FRONT END) ആണ്.

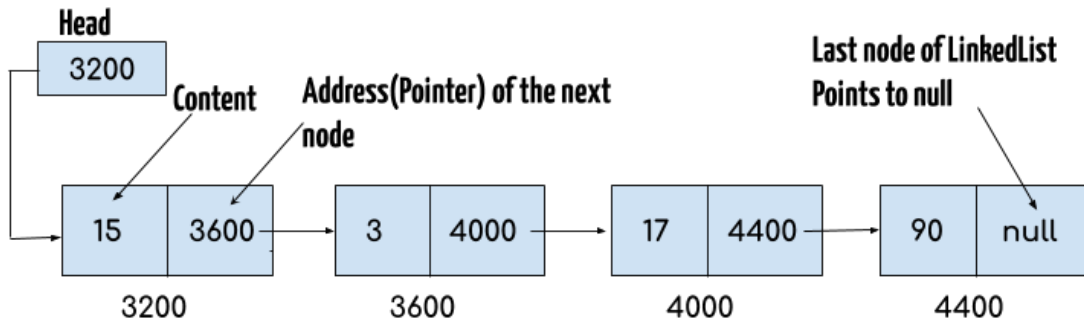
- It follows FIFO(First In First Out) principle.
ഇവിടെ ആദ്യം add ചെയ്യുന്ന എലമെന്റ് ആണ്, ആദ്യം ഡിലീറ്റ് ചെയ്യാൻ സാധിക്കുക (**FIFO**)



LINKED LIST (ലിങ്ക്ഡ് ലിസ്റ്റ്)

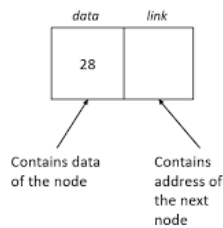


Click or Scan here to see the Video



- Linked list is a collection of node.

പരസ്പരം കൂട്ടി ചേർത്ത നോഡുകളുടെ ശേഖരമാണിത്.



- Each node consists of two parts – a data and a link (Contains the address of the next node)

ഓരോ നോഡിലും രണ്ട് ഭാഗങ്ങളുണ്ട് - ഒരു ഡാറ്റയും ഒരു ലിങ്കും (ലിങ്കിൽ അടുത്ത നോഡിന്റെ വിലാസം അടങ്ങിയിരിക്കുന്നു)

- The address of the first node is stored in a special pointer called START.

ആദ്യ നോഡിന്റെ വിലാസം START എന്ന പ്രത്യേക പോയിന്ററിൽ സംഭരിച്ചിരിക്കുന്നു.

- The address of the Last node is stored with NULL.

അവസാന നോഡിന്റെ വിലാസമായി NULL (ശൂന്യം) എന്ന value സൂക്ഷിച്ചിരിക്കുന്നു.

