Introduction to Data Structure

Rakesh Amrutkar
Under the guidance
of
Prof. Pranav Nerurkar

V.J.T.I

February 13, 2019

Outline

1. Introduction to Data structure
2. Need of data structure
3. Types of data structure
4. Stack
5. Queue
6. Linked list
7. Types of Linked list
8. Basic operation on linked list

Introduction to Data Structure

- In computer science, a data structure is a data organization, management and storage format that enables efficient access and modification.
- Data structure is a collection of data values, the relationships among them, and the functions or operations that can be applied to the data.

Examples

Stack , Queue ,Tree etc.

Need of data Structure

Problems :

- Processor speed: To handle very large amout of data, high speed processing is required, but as the data is growing day by day to the billions of files per entity, processor may fail to deal with that much amount of data.
- Data Search: Consider an inventory size of 106 items in a store, If our application needs to search for a particular item, it needs to traverse 106 items every time, results in slowing down the search process.

Types of data Structure

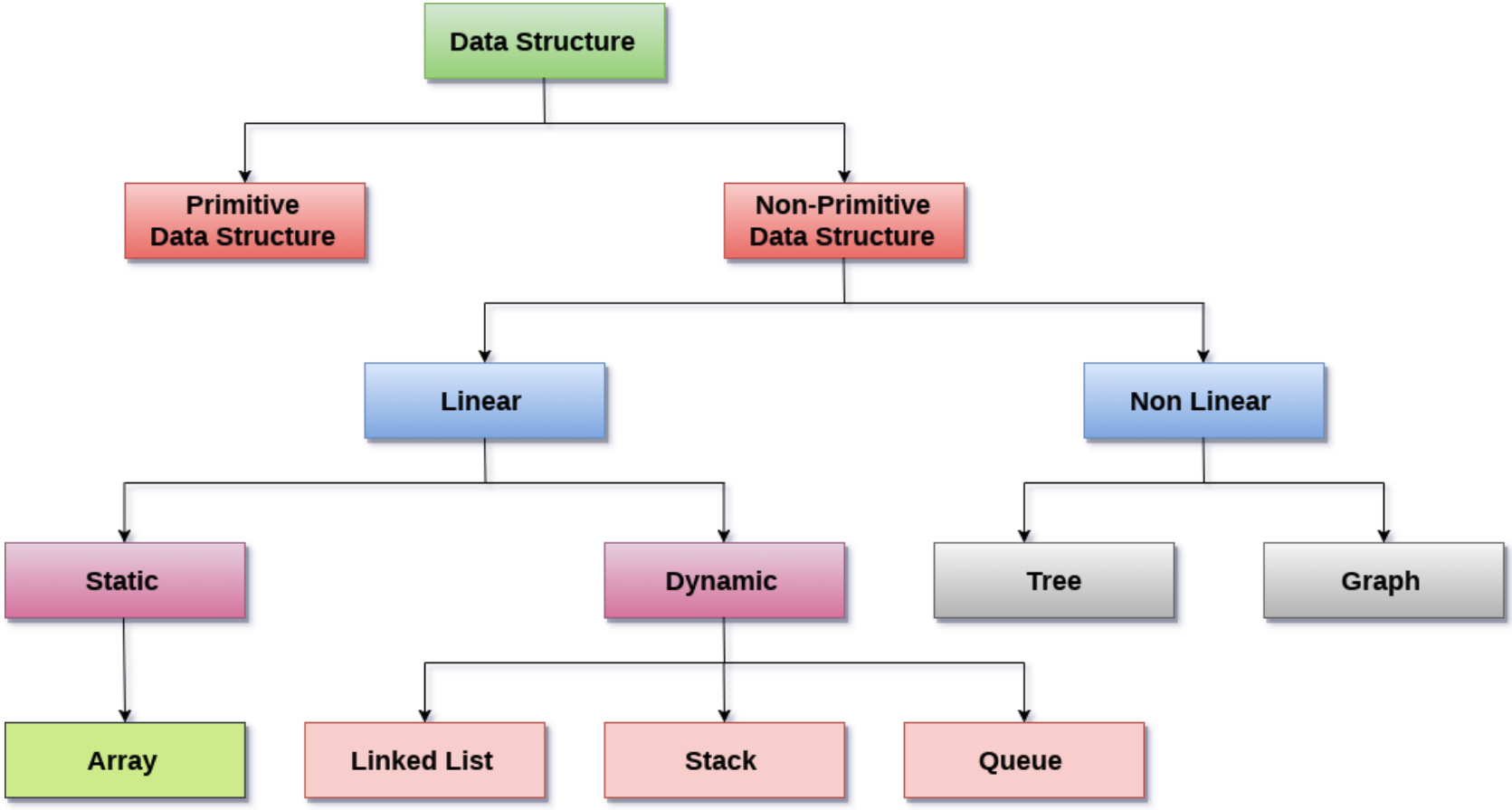


Figure 1: Types of Data Structure

Types of data Structure

- Array
- Stack
- Queue
- Linked list
- Tree
- Binary Search tree
- Graph

Stack

- Stack is an ordered list of similar data type.
- Stack is a LIFO(Last in First out) structure or we can say FILO(First in Last out).
- push() function is used to insert new elements into the Stack and pop() function is used to remove an element from the stack. Both insertion and removal are allowed at only one end of Stack called Top.
- Stack is said to be in Overflow state when it is completely full and is said to be in Underflow state if it is completely empty.

Stack

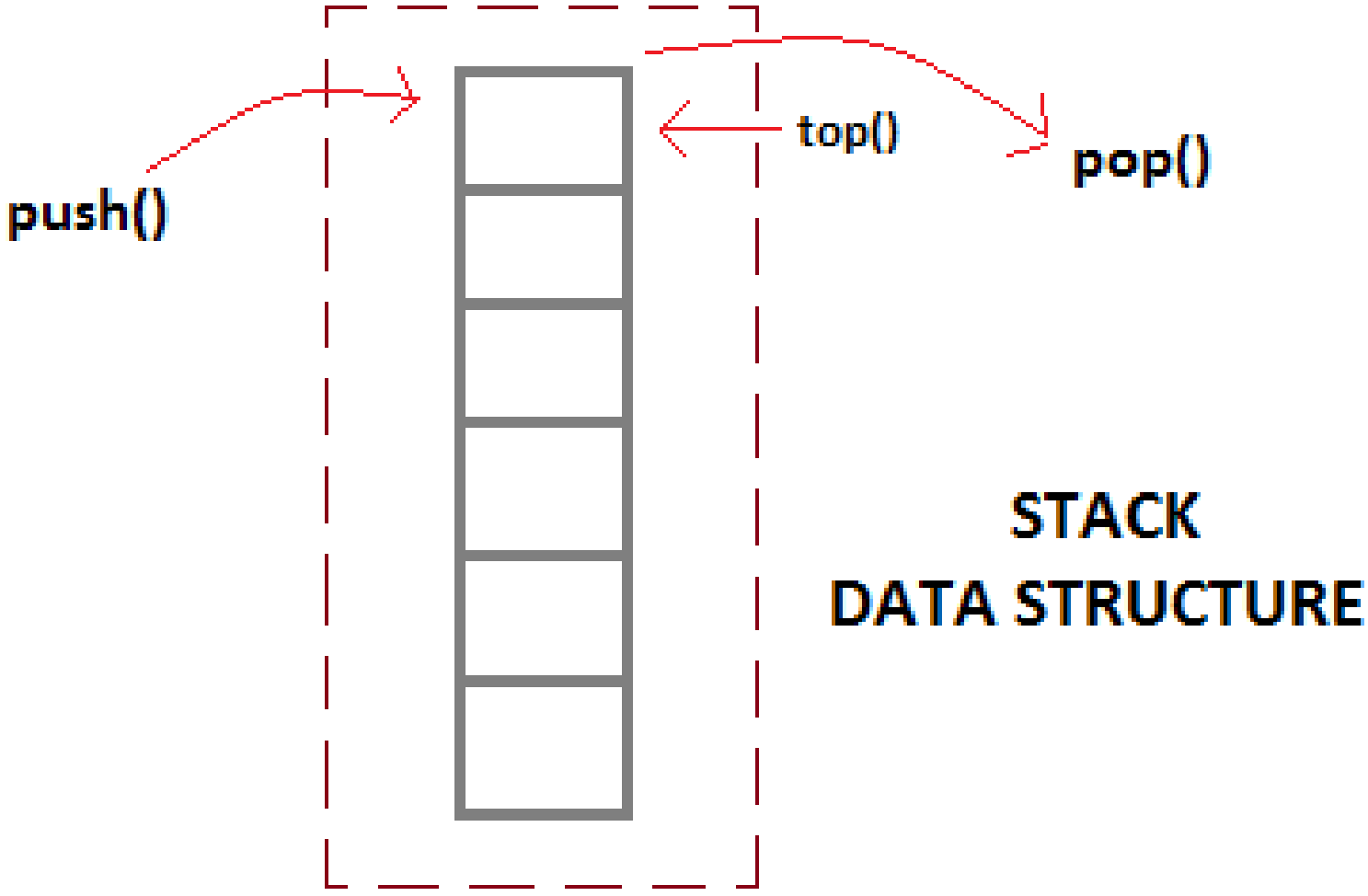
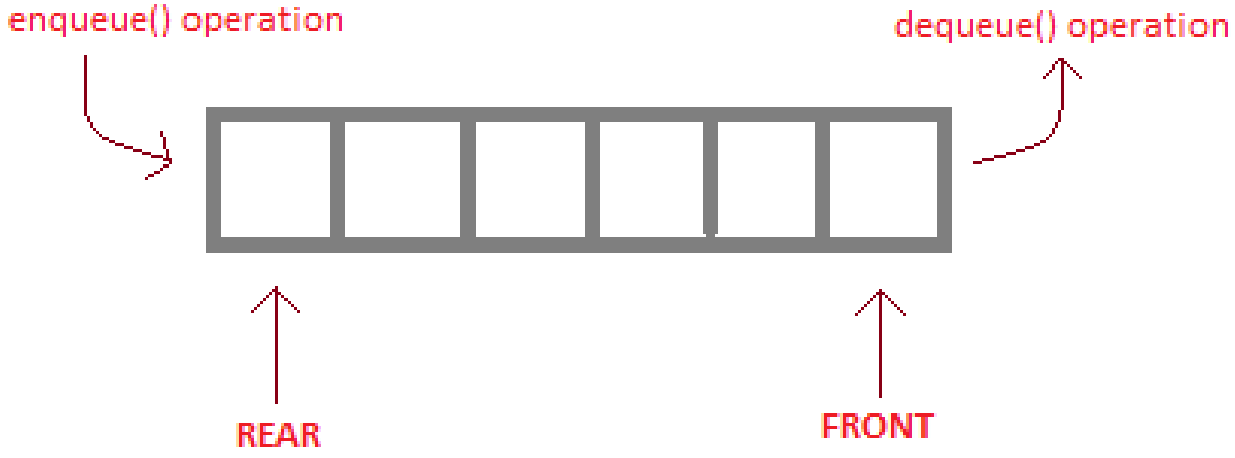


Figure 2: Stack Operation

Queue

- A Queue is a linear structure which follows a particular order in which the operations are performed.
- The order is First In First Out (FIFO).
- A good example of a queue is any queue of consumers for a resource where the consumer that came first is served first.
- In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added.

Queue



enqueue( ) is the operation for adding an element into Queue.

dequeue( ) is the operation for removing an element from Queue .

QUEUE DATA STRUCTURE

Figure 3: Queue operations

Linked list

- Linked List is a linear data structure. Unlike arrays, linked list elements are not stored at contiguous location; the elements are linked using pointers.
- A linked list is represented by a pointer to the first node of the linked list. The first node is called head. If the linked list is empty, then value of head is NULL.
- Each node in a list consists of at least two parts:
  1) Data
  2) Pointer (Or Reference) to the next node

Linked list

Representation of Linked List:

```
struct Node
{
int data;
struct Node *next;
};
```

Types of Linked List

Following are the various types of linked list.

- Simple Linked List – Item navigation is forward only.
- Doubly Linked List – Items can be navigated forward and backward.
- Circular Linked List – Last item contains link of the first element as next and the first element has a link to the last element as previous.

Basic Operations

Following are the basic operations supported by a list.

- Insertion – Adds an element at the beginning of the list.
- Deletion – Deletes an element at the beginning of the list.
- Display – Displays the complete list.
- Search – Searches an element using the given key.
- Delete – Deletes an element using the given key.

Insertion operation

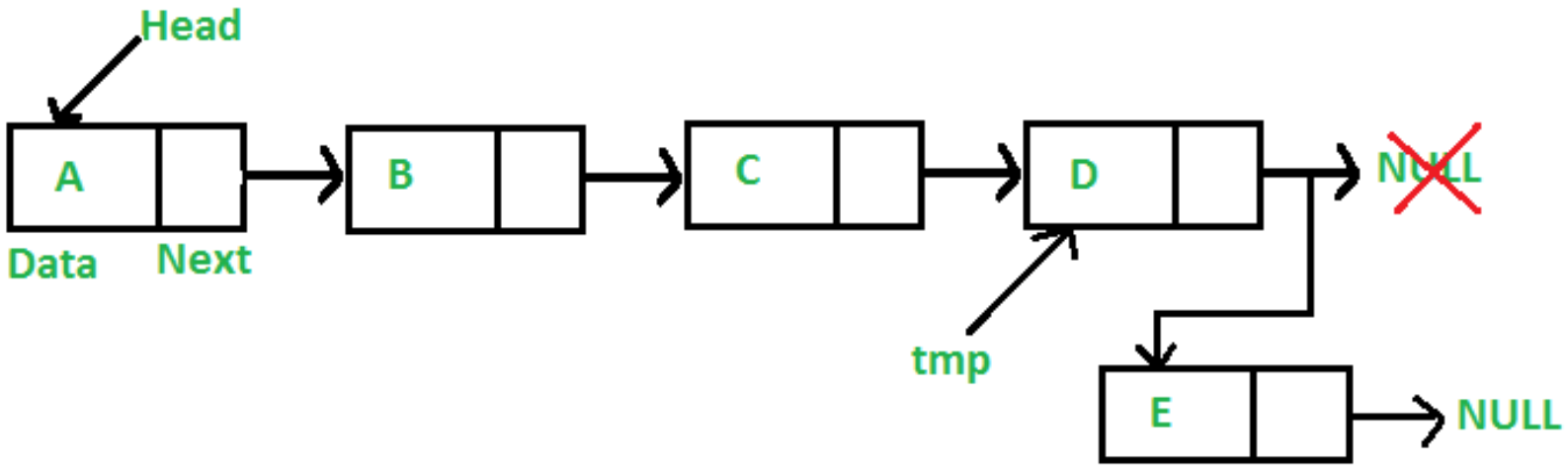


Figure 4: Insertion of a new node at the end of a linked list

Insertion

```
void append(struct Node** headref, int newdata)
{
struct Node* newnode = (struct Node*) malloc(sizeof(struct
Node));
struct Node *last = *headref;

newnode->data = newdata;

newnode->next = NULL;

if (*headref == NULL)
{ *headref = newnode;
}

while (last->next != NULL)
  last = last->next;

last->next = newnode;
}
```

References

1. www.geeksforgeeks.com

2. www.onlinetutorialspoint.com

THANK YOU