

Data Structures: Linear_List

Lists

Linear

Linear List

- Linear list stores a collection of objects of a certain type, denoted as elements of a list.
 - Their two types of linear lists.
 - Stack
 - Queue
-

Stack

- A stack is generally a data structure that operates on the last in and first out principle. it is an abstract data type.
- This means all data operations can be carried out only on the last in first out principle.
- The push and pop operations are used for inserting and deleting an element from a stack.
- A stack can be implement a stack using a list.

```
options(tinytex.verbose = TRUE)

stack_list <- list(1,2,3,5,7)

print('Examples are in Google Colab')
```

Stack example

```
## [1] "Examples are in Google Colab"
```

Queue

- A queue is a linear type of data structure used to store the data in a sequentially. The concept of queue is based on the FIFO, which means “First in First Out”. It is also known as “first come first served”. The queue has the two ends front and rear. The next element is inserted from the rear end and removed from the front end.
- We can perform the following operations in the Queue:
 - Enqueue - The enqueue is an operation where we add items to the queue.
 - Dequeue - The dequeue is an operation where we remove an element from the queue. An element is removed in the same order as it is inserted. If the queue is empty, it is a condition of the Underflow.
 - Front - An element is inserted in the front end.

- Rear - An element is removed from the rear end.

```
print('Examples are in Google Colab')
```

Queue example in R

```
## [1] "Examples are in Google Colab"
```