GROUP 4

CYCLIC DOUBLY LINKED LIST

Alyssa Hardy | Bryce Hankinson | Grant Solomon | Samuel Davis

CYCLIC DOUBLY LINKED LIST

- Contains properties of both a doubly linked list and a cyclic linked list
- A linked list is a linear data structure that links consecutive nodes
- The nodes in a doubly linked list have pointers to both the next and previous node in the list
- The first and last nodes of a doubly linked list are immediately accessible and therefore allow traversal of the list from the beginning or end of the list
- Any node, once attained, can be used to begin a new traversal of the list, in either direction from the given node

NODE

- Every node contains three fields (pictured right):
  - A pointer to the previous node
  - A pointer to the next node
  - The data the node possesses

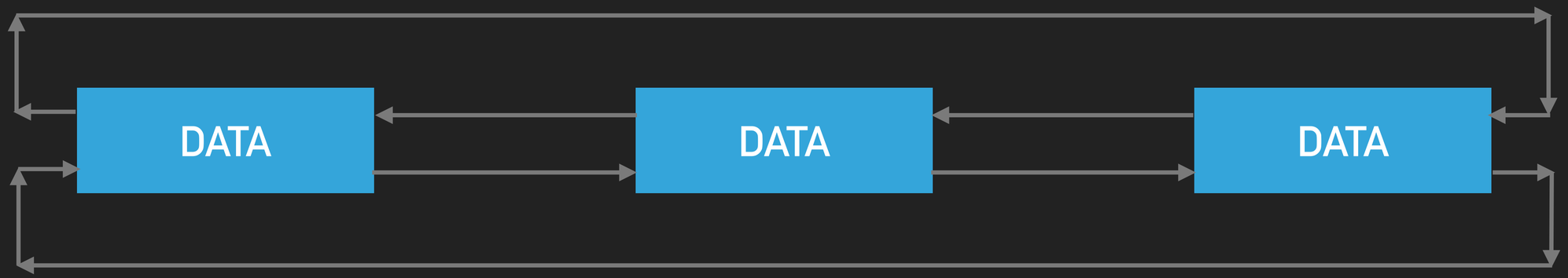
```
class Node<E> {
    private E element;
    private Node<E> next;
    private Node<E> previous;

    public Node(E e, Node<E> next, Node<E> previous, int index){
        element = e;
        this.next = next;
        this.previous = previous;
    }
```

- In a Cyclic Doubly Linked List, the first node points to the last node with its previous pointer and the last node points to the first node with its next pointer.

CYCLIC DOUBLY LINKED LIST

- In a Cyclic Doubly Linked List, the first node points to the last node with its previous pointer and the last node points to the first node with its next pointer.



ADVANTAGES OF A CYCLIC DOUBLY LINKED LIST

- Can be traversed from head to tail or vice versa
- Like a normal linked list, the size is dynamic
- Nodes can be added or deleted to the beginning/end of the list in constant time O(1)
- The circular nature of the list allows for nodes to easily be visited more than once