ECE 250 - Project 1
Deque as Linked List
Design Document

Ayushi Tiwari, UW UserID: attiwari

Jan 31st, 2020

1. Overview of Classes

Class Node

Definition: Represents a generic node for the double linked list. Each node is a reference to the address of the data and also defines the next and previous nodes.

Member Variables

1. data (data type: Int)	*stores the value to be added*
2. next (data type: Node type pointer)	*defines the next node*
3. previous (data type: Node type pointer)	*defines the last node*

Class Deque

Definition: Deque is a type of queue data structure implemented using doubly linked list. The class stores the data in the list using Node class, inserts and deletes at the front and end of the linked list, compares the first and last node with an integer, prints and clears the list, etc.

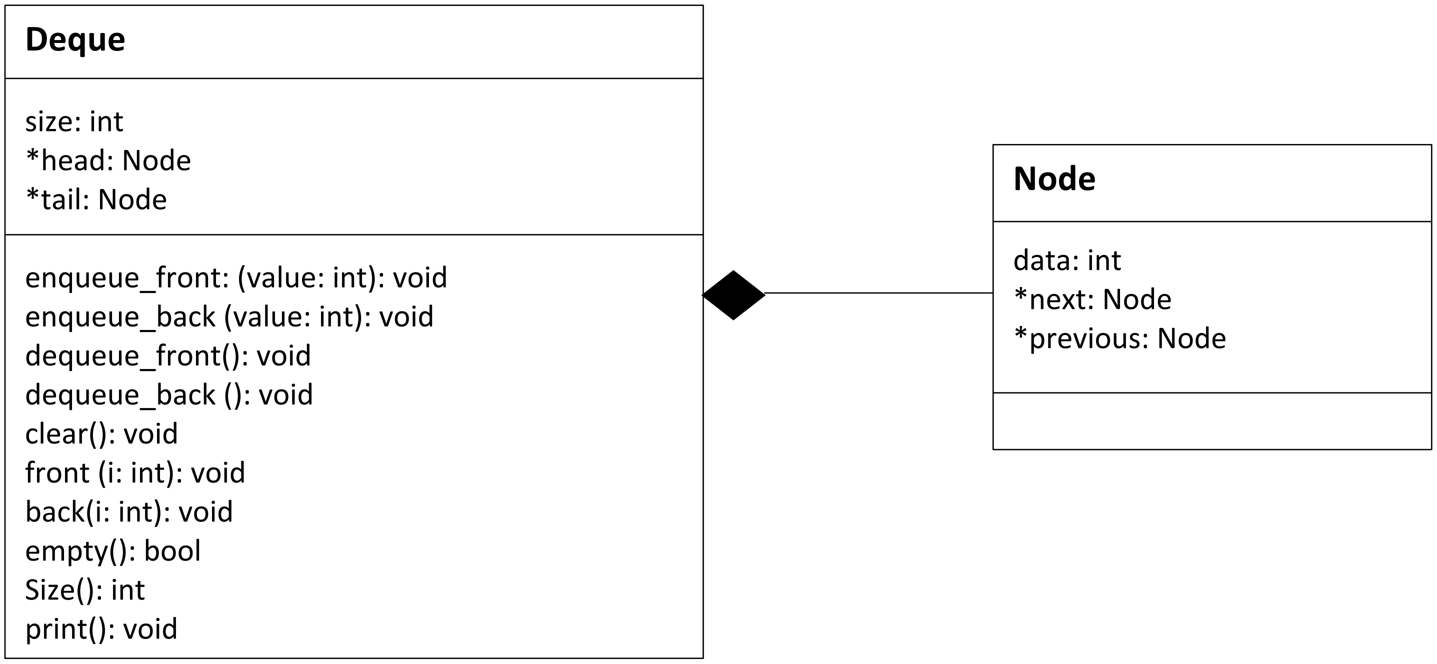
Member Variables

1. head (data type: Node type pointer)	*Address of the first element of the list*
2. tail (data type: Node type pointer)	*Address of the first element of the list*
3. size (data type: Int)	*Size of the Deque list*

Member Functions

1. enqueue_front: Adds data in the new node created at the front and increments size
2. enqueue_back: Adds data in the new node created at the end and increments size
3. dequeue_front: Deletes the node at the front and shifts the list and decrements size
4. dequeue_back: Deletes the node at the end and shifts the list and decrements size
5. clear: deleted the deque list
6. front: Compares the data of the head node with an integer
7. back: Compares the data of the end node with an integer
8. empty: checks if the list is empty
9. Size: returns the size of the list
10. print: prints the deque list twice: from front to back and vice versa

2. UML Class Diagram



3. Constructors/ Destructors

Class Node:

1. Constructor: Initially assigns the data as 0, next and previous pointer as nullptr
2. Destructor: It frees the memory used for creating the node by deleting next and previous pointers

Class Deque:

1. Constructor: Creates an empty linked list by default
2. Destructor: Frees the memory space by deleting head and tail pointers

3. Test Cases

Test 1: Data is added to the front and end node
Test 2: Deletes the data at the frond and end
Test 3: Add some nodes in the list and compare the front and end and delete the list
Test 4: Check the size after adding and deleting nodes

Test1 Example

```
enqueue_front 2
enqueue_back 2
front 1
back 10
back 2
print
deque_front
front 3
empty
print
clear
clear
dequeue_back
empty
enqueue_front 4
print
```

Test2 Example

```
empty
enqueue_back 1
front 1
back 1
print
enqueue_bacl 2
enqueue_back 3
fron 2
print
```

4. Performance

Deque class is a doubly linked list which can insert, delete and find the value for first and last nodes in O(1) time.
For print and clear function, it goes through all the nodes in O(n) time.

5. Sources

For the source code:
https://www.softwaretestinghelp.com/doubly-linked-list/
https://www.hackerearth.com/practice/notes/doubly-linked-list-data-structure-in-c/
https://www.geeksforgeeks.org/doubly-linked-list/
https://www.geeksforgeeks.org/implementation-deque-using-doubly-linked-list/

For the test cases:
https://github.com/ece23/ECE250-testCases/tree/master/p1