**import** java.util.NoSuchElementException;

**public** **class** Queue {

**public** **class** Node {

**int** key;

Node next;

Node prev;

**public** Node(**int** key, Node next, Node prev){

**this**.key = key;

**this**.next = next;

**this**.prev = prev;

}

}

**public** **static** **int** *size*;

**static** Node *head*;

**public** Queue(){ // constructor

*size* = 0;

*head* = **null**;

}

**public** **void** add(**int** a){

**if**(*size* == 0){

Node temp = **new** Node(a,**null**, **null**);

*head* = temp;

*size*++;

}

**else**{

Node ptr = *head*;

**while**(ptr.next!=**null**){

ptr = ptr.next;

}

Node temp = **new** Node(a,**null**,ptr);

ptr.next = temp;

*size*++;

}

}

**public** **int** remove()

**throws** NoSuchElementException{

**if**(*size* == 0){

**throw** **new** NoSuchElementException("The Data Structure is Empty!");

}

**else** **if**(*size* == 1){

**int** r = *head*.key;

*head* = **null**;

*size*--;

**return** r;

}

**else**{

**int** temp = *head*.key;

*head* = *head*.next;

*head*.prev = **null**;

*size*--;

**return** temp;

}

}

**public** **static** **int** getValue(**int** i){

**int** value = -1;

**if** (i > *size*){

**return** value;

}

Node ptr =*head*;

**int** count = 0;

**while**(ptr !=**null**){

**if**(count == i){

**break**;

}

count++;

ptr = ptr.next;

}

value = ptr.key;

**return** value;

}

**private** **static** **void** traverse(){

Node ptr = *head*;

System.***out***.println("The data structure is: ");

**while**(ptr != **null**){

System.***out***.print(ptr.key + " ");

ptr = ptr.next;

}

System.***out***.println();

System.***out***.println();

}

**public** **static** **void** main(String[] args){

Queue data = **new** Queue();

data.add(1);

data.add(9);

data.add(4);

data.add(5);

data.add(10);

data.add(0);

*traverse*();

**int** get0 = *getValue*(0);

**int** get3 = *getValue*(3);

System.***out***.println("getValue(0) is: "+ get0);

System.***out***.println("getValue(3) is: "+ get3);

}

}

1.

**int** temp = *head*.key;

*head* = *head*.next;

*head*.prev = **null**;

*size*--;

**return** temp;

Remove only delete the first item in my data structure. My data structure is doubly linked list. So every time delete, I only need to delete the first node. O(1)

**while**(ptr !=**null**){

**if**(count == i){

**break**;

}

count++;

ptr = ptr.next;

}

getValue I only compared for i times i is a constant. So O(i) ~= O(1)

**public** **void** add(**int** a){

**if**(*size* == 0){

Node temp = **new** Node(a,**null**, **null**);

*head* = temp;

*size*++;

}

**else**{

Node ptr = *head*;

**while**(ptr.next!=**null**){

ptr = ptr.next;

}

Node temp = **new** Node(a,**null**,ptr);

ptr.next = temp;

*size*++;

}

}

Because this is a doubly linked and first node is first element in the data structure. Every time I need to add at end. So O(n)