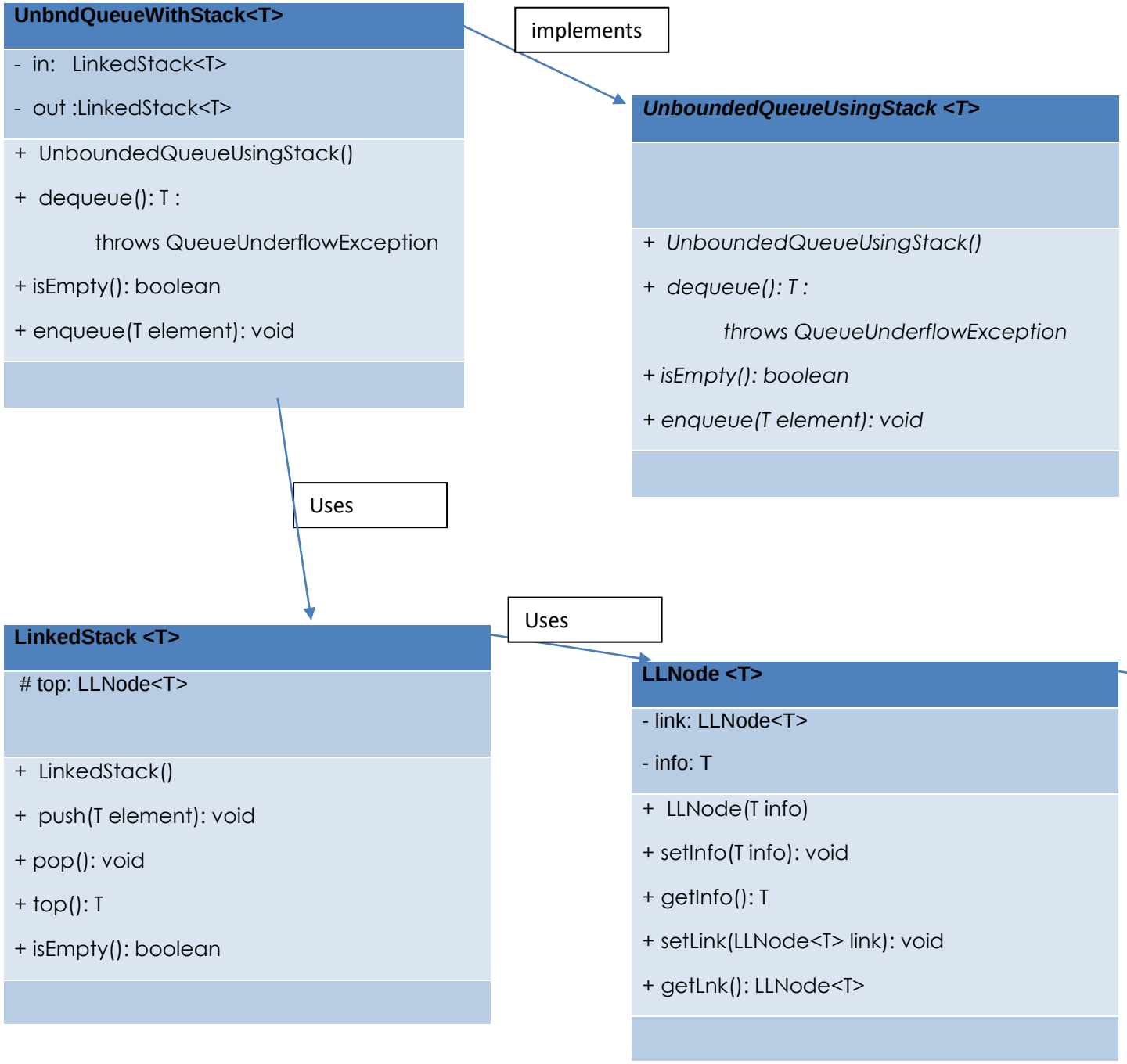
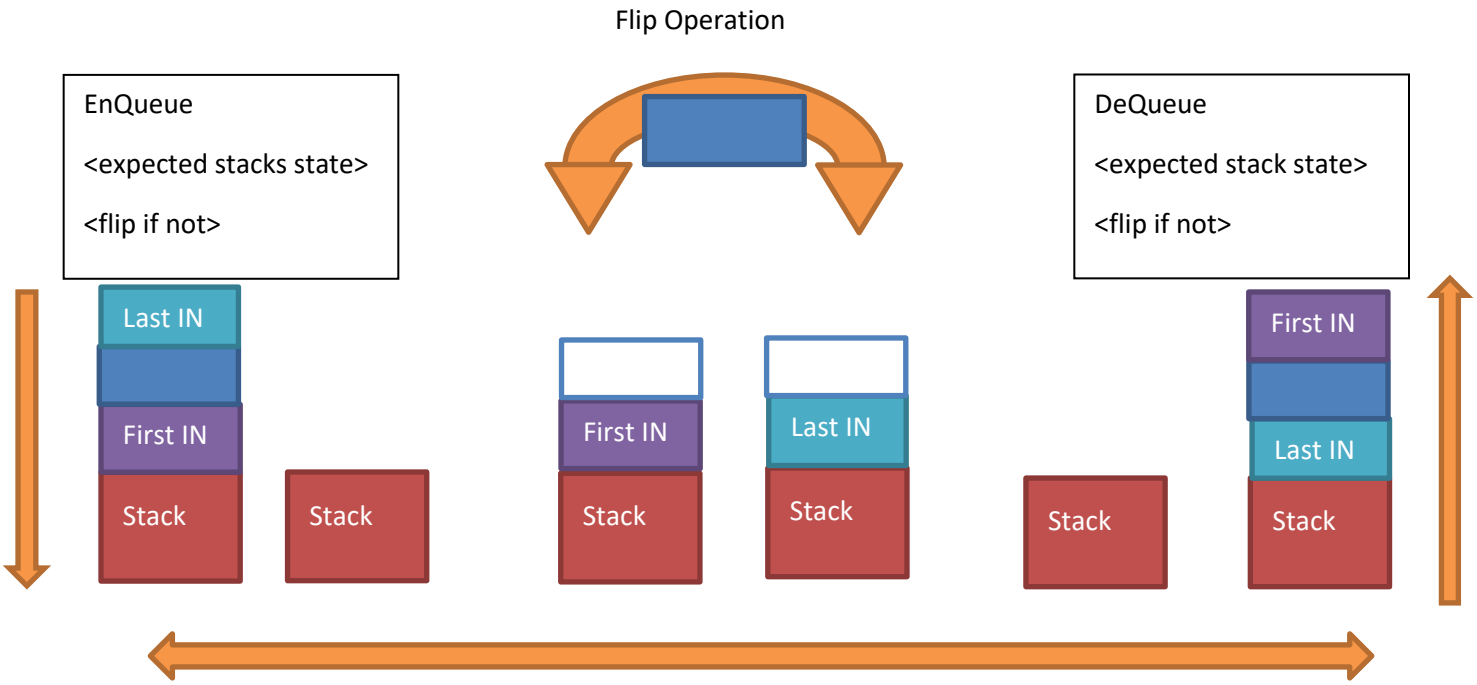

UnbndQueueWithStack<T>
- in: LinkedStack<T>
- out :LinkedStack<T>
+ UnboundedQueueUsingStack()
+ dequeue(): T :
throws QueueUnderflowException
+ isEmpty(): boolean
+ enqueue(T element): void
implements
UnboundedQueueUsingStack <T>
+ UnboundedQueueUsingStack()
+ dequeue(): T :
throws QueueUnderflowException
+ isEmpty(): boolean
+ enqueue(T element): void
Uses
LinkedStack <T>
top: LLNode<T>
+ LinkedStack()
+ push(T element): void
+ pop(): void
+ top(): T
+ isEmpty(): boolean
Uses
LLNode <T>
- link: LLNode<T>
- info: T
+ LLNode(T info)
+ setInfo(T info): void
+ getInfo(): T
+ setLink(LLNode<T> link): void
+ getLnk(): LLNode<T>

Stack is a First in, Last Out,
Queue is First in, First Out

Considering the above, you can move the elements from stack 1, to stack 2, effectively the bottom element will become the top, which than can be the first element removed while also being the first element added. Basically from the abstract perspective, its first in, first out with the following steps taken its Enqueue (First in, last out) -> Flip operation (first becomes last, and last becomes first) -> Dequeue-> (Flip back) ->Enqueue again)



moving elements around to flip the contents to either remove elements or add elements (the side must be kept track by some means, in my case, I do a while loop that move elements from one stack to another, checking for isEmpty, if a stack is empty, the while loop ends (the while loop may never run if elements already in the correct side))

Implementation thoughts

LinkedStack used as the stacks instead of ArrayUnbndQueue

I figured using a linked stack would be a better idea, when the ArrayUnbndstack is increased in size, eventually both stacks will need to expand in size causing a O(2n) on both enqueue and dequeue operation (eventually). Another issue with ArrayUnbndStack is that 50% of the slots or more will be empty after any flip operation, wasting memory space and having to expand two arrays, the operation just seems very wasteful (even more so than the double stack idea). Instead I see it technically better from an implementation standpoint to use a linked stacked since its memory will be recycled. however, it might be more efficient if the array are within expected sizes to use two array stacks, however since it's an unbndqueue, I would say predicting the size accurately is in-probable