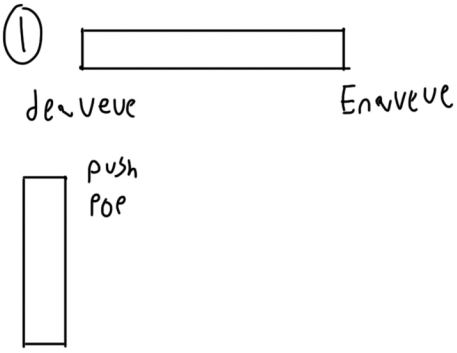
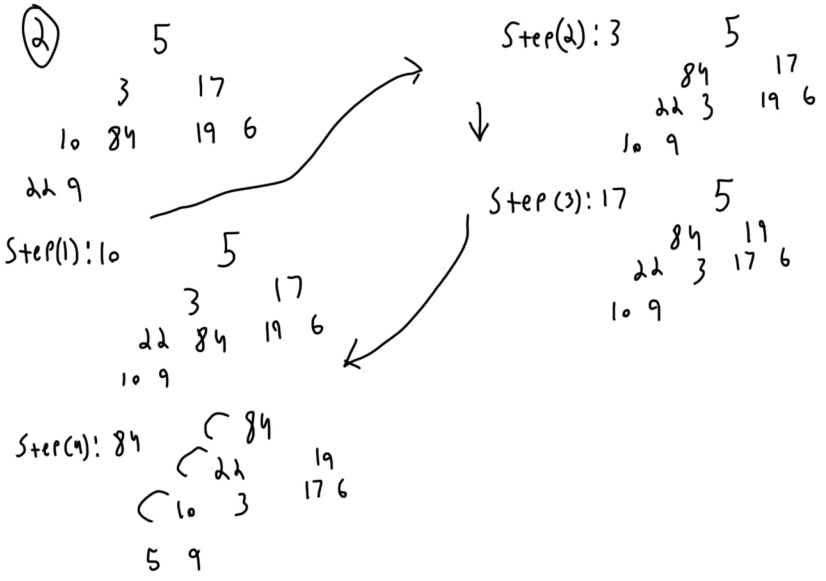
① 


```
S.Push(item)
    Enqueue
```

S.push() run time is constantine, becuase we just have to place the next item at the top using enqueue. O(1).

```
S.Pop()
    if (q.size() == 0)
        return ("stack is empty")
    else:
        Initialize new queue
        For(i=0; i<q.size-1; i++)
            Newqueue.enqueue(dequeue)
        Old queue = Newqueue.
```

Time complexity would be 2n, because for each element in the queue we have to dequeue and enqueue except the last one. O(2n-1)

② 

③ To implement a fifo priority queue, assign each item inserted a counter based on when it was inserted. item1 = 0, item2 = 1, item3 = 2....

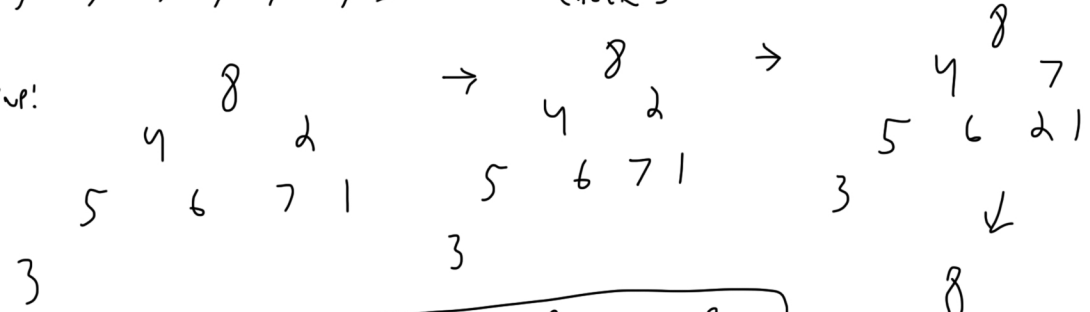
```
Enqueue(item)                    - Sudo
    Push item(Counter, item)
    Counter += 1
```

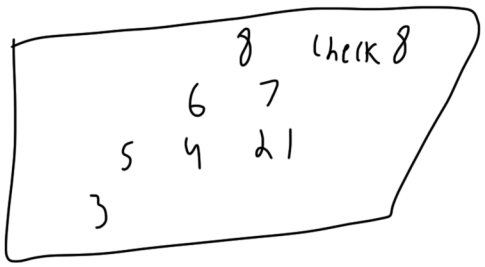
When dequeued, the item with the smallest counter will be removed first. This means first item in is first to leave.

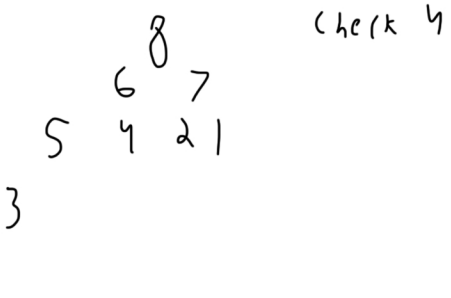
To implement a stack, we can make our counter negative. Therefore, as we insert items, the counter gets smaller. Since we dequeue the item with the smallest counter, that means that the last item in, is the first to be removed.

[8, 4, 2, 5, 6, 7, 1, 3]

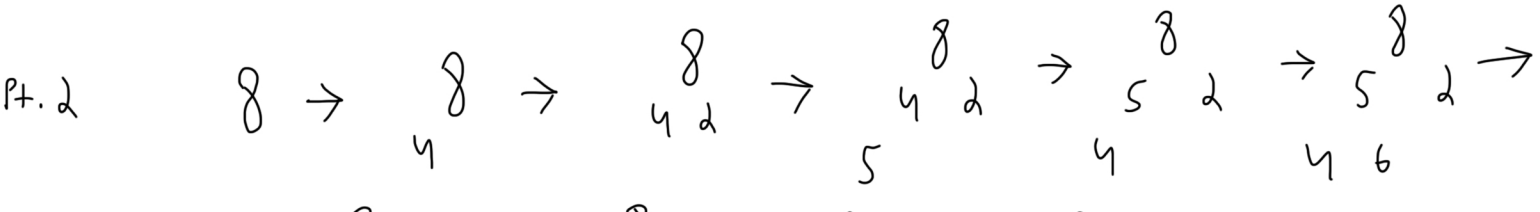
build heap:

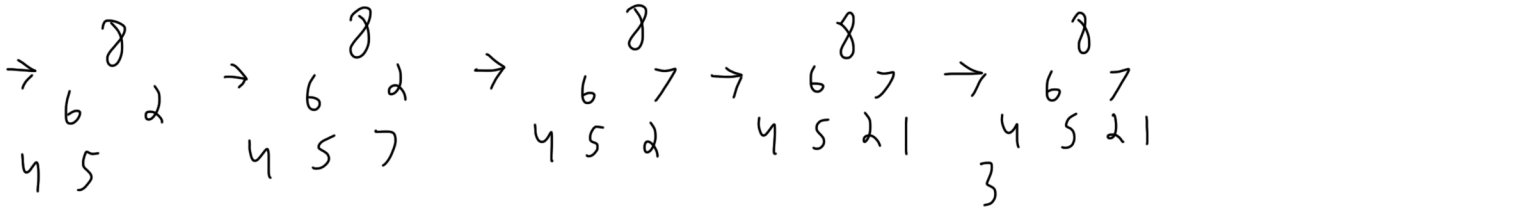


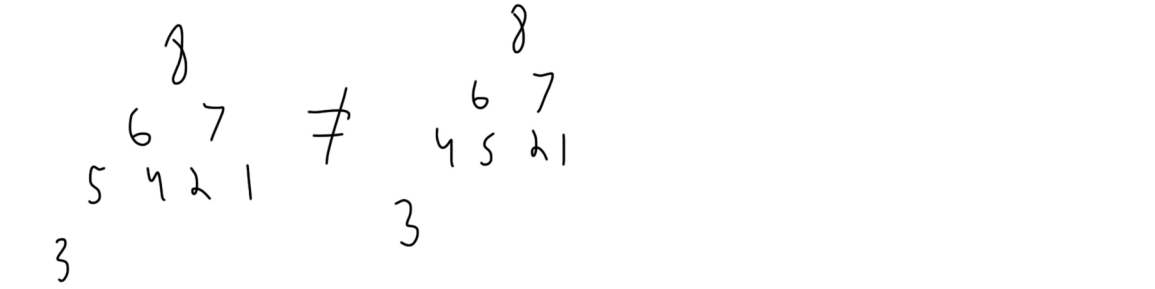




Pt. 2







8 6 7 5 4 2 1 3

8 6 7 4 5 2 1 3