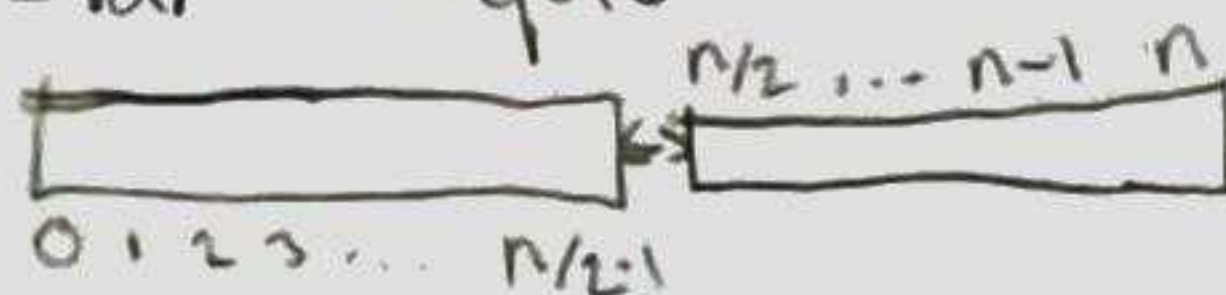


$$O(1 + \min\{i, n-i, \frac{n}{2} - i\})$$

$$= O(1+i) \text{ or } O(1+n-i) \text{ or } O(\frac{n}{2}-i)$$

Our Treque will be 2 arrays?



This data structure is similar to a deque but also has the benefit that adding/removing from the middle will have a runtime of $\approx O(1)$ or constant time. It gets the native benefits of the deque for free like constant time add/remove from the front & back.

Get & set will be constant time as well because arrays benefit is their 'Random access' to any element with constant time.

The downside of this structures runtimes are due to it maintaining order in the array. Unlike a data type where order is not important if we want to add/remove from the first $\frac{n}{4}$ elements it will take $O(1+i)$, if we want to add/remove from $\frac{2n}{4} \rightarrow \frac{3n}{4}$ it will take $O(\frac{n}{2}-i)$, and if you want to add/remove from the last quarter of the elements $O(1+(n-i))$ would be your runtime.

<p>Get(i);</p> <p>if $i > \text{front.size} - 1$ $\text{back.get}(i - \text{front.size})$ else $\text{front.get}(i)$</p>	<p>Set(i, x);</p> <p>if $i > \text{front.size} - 1$ $\text{back.set}(i - \text{front.size}, x)$ else $\text{front.set}(i, x)$</p>	<p>* in my implementation I flipped this if statement structure ie:</p> <div style="border: 1px solid black; padding: 5px;"> <p>"if $i < \text{front.size}$ $\text{front.set}(i, x)$ else $\text{back.set}(i - \text{front.size}, x)$ rebalance"</p> </div> <p>this doesn't change much except I don't need the "-1" in the "if" condition.</p>
<p>Add(i, x);</p> <p>if $i > \text{front.size} - 1$ $\text{back.add}(i - \text{front.size}, x)$ else $\text{front.add}(i, x)$</p> <p>rebalance;</p>	<p>Remove(i);</p> <p>if $i > \text{front.size} - 1$ $\text{back.remove}(i - \text{front.size})$ else $\text{front.remove}(i)$</p> <p>rebalance;</p>	

function to rebalance
front & back