Sorting Hocky Pucks Problem

The variables used in this solution are split into 3 parts. The start, the buffer, and the exit.

```
Queue start = new Queue();
Stack buffer = new Stack();
Queue exit = new Queue();

for (Integer item: arr) {
    start.push(item);
}
```

The start array, and the exit are of type queue, as the pucks should go in a First In First Out (FIFO) order. The buffer is made into a stack since it requires the item to go in a specific order (First In Last Out - FILO).

With the help of a counter that keep track of what the value of the next element should be, a while loop is used to iterate until the start and buffer are empty.

```
while(!start.isEmpty() || !buffer.isEmpty()){ }
```

Through every iteration, we check if the beginning of the queue matches the counter and if so we move the item to the exit.

```
if (start.peek()!=null && start.peek()==counter) {
    exit.push(start.pop());
    counter++;}
```

Only if the start doesn't match the counter, we check if the buffer matches the counter.

```
else if(buffer.peek()!=null && buffer.peek()==counter) {
    exit.push(buffer.pop());
    counter++;}
```

If the buffer doesn't match the counter and there's an item in the start we move it to the buffer. Otherwise we just exit the loop and return the size of the exit queue, because we can't move anything to the exit or the buffer.

```
else if (start.peek()!=null){
    buffer.push(start.pop());
}else {break;}
```