```
if( arr.length == 0 || arr.length == 1 )
    sorted = true;
```

n = 7 arr = [3, 4, 7, 5, 6, 2, 1] int i = 0 // use to iterate thru arr L→R

	0	1	2	3	4	5	6
1	3	4	7	5	6	2	1
2	3	4	7	5	6	2	1
3	3	4	5	7	6	2	1
4	3	4	5	6	7	2	1
5	2	3	4	5	6	7	1
6	1	2	3	4	5	6	7

i=0

1: arr[i+1] < arr[i] false => i++ // i=1

2: arr[i+1] < arr[i] false => i++ // i=2

3: arr[i+1] < arr[i] true swap*

   arr[i] < arr[i-1] false => i++ // i=3

4: arr[i+1] < arr[i] true swap*

   arr[i] < arr[i-1] false => i++ // i=4

5: arr[i+1] < arr[i] true swap*

   arr[i] < arr[i-1] true swap*

   arr[i] < arr[i-1] true swap*

   arr[i] < arr[i-1] true swap*

   arr[i] < arr[i-1] must deal will out of bounds exception (null)

1: arr[i+1] < arr[i] false i++ // i=1

2: arr[i+1] < arr[i] false i++; i=2

3: arr[i+1] < arr[i] true

   arr[3] < arr[2]

   3 4 7 5 6 2 1

```
swap(arr[3], arr[2], arr)
tmp = arr[hi]
arr[hi] = arr[lo];
```

i							
	0	1	2	3	4	5	6
i=0	3	4	7	5	6	2	1
i=1	3	4	7	5	6	2	1
i=2	3	4	7	5	6	2	1
i=3	3	4	5	7	6	2	1
i=4	3	4	5	6	7	2	1
i=5	2	3	4	5	6	7	1
i=6	1	2	3	4	5	6	7

n=7

	0	1	2	3	4	5	6	
i=0	3	4	7	5	6	2	1	
i=1	3	4	7	5	6	2	1	
i=2	3	4	5	7	6	2	1	
i=3	3	4	5	6	7	2	1	
i=4	2	3	4	5	6	7	1	
i=5	1	2	3	4	5	6	7	

	0	1	2	3	4	5	6	
i=0	3	4	7	5	6	2	1	Red - swap
i=1	3	4	7	5	6	2	1	Blue - Backtracking swaps
i=2	3	4	7	5	6	2	1	swap Green - i
	3	4	5	7	6	2	1	no swaps
i=3	3	4	5	7	6	2	1	swap
	3	4	5	6	7	2	1	no swap
i=4	3	4	5	6	7	2	1	
	3	4	5	6	2	7	1	
	3	4	5	2	6	7	1	
	3	4	2	5	6	7	1	
	3	2	4	5	6	7	1	
	2	3	4	5	6	7	1	* Must deal w/ out of Bound Except.
i=5	2	3	4	5	6	7	1	
	2	3	4	5	6	1	7	
	2	3	4	5	1	6	7	
	2	3	4	1	5	6	7	
	2	3	1	4	5	6	7	
	2	1	3	4	5	6	7	
	1	2	3	4	5	6	7	* out of Bound Except

n=6

	0	1	2	3	4	5	
i=0	1	4	3	5	6	2	
i=1	1	3	4	5	6	2	
i=2	1	3	4	5	6	2	
i=3	1	3	4	5	6	2	
i=4	1	2	3	4	5	6	

	0	1	2	3	4	5	
i=0	1	4	3	5	6	2	
i=1	1	4	3	5	6	2	
	1	3	4	5	6	2	
i=2	1	3	4	5	6	2	
i=3	1	3	4	5	6	2	
i=4	1	3	4	5	6	2	
	1	3	4	5	2	6	
	1	3	4	2	5	6	
	1	3	2	4	5	6	
	1	2	3	4	5	6	