Bubble Sort
Bubble Sort

What do we do first?
Bubble Sort

```
0 1 2 3 4 5
10 4 3 0 11 8
```

```
j - 1 j
0 0
```

Compare j-1 and j; Swap if L[j-1] > L[j]

What next?
Bubble Sort

len = 6

Compare j-1 and j; Swap if L[j-1] > L[j]

What next?
Bubble Sort

len = 6

```
0 1 2 3 4 5
4 10 3 0 11 8
```

Compare j-1 and j; Swap if L[j-1] > L[j]

What next?
Bubble Sort

len = 6

0 1 2 3 4 5

4 3 10 0 11 8

Compare j-1 and j; Swap if L[j-1] > L[j]

What next?
Bubble Sort

len = 6

Compare $j-1$ and $j$; Swap if $L[j-1] > L[j]$

What next?
Bubble Sort

len = 6

```
[4, 3, 0, 10, 11, 8]
```

Compare j-1 and j; Swap if L[j-1] > L[j]

What next?
Bubble Sort

```
4  3  0  10  11  8
```

len = 6

Compare j-1 and j; Swap if L[j-1] > L[j]

What next?
Bubble Sort

```
0  1  2  3  4  5
4  3  0  10 11 8
```

What next?

**Compare** j-1 **and** j; **Swap** if L[j-1] > L[j]

len = 6
Bubble Sort

Compare j-1 and j; Swap if L[j-1] > L[j]

What next?
**Bubble Sort**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>10</td>
<td>8</td>
<td>11</td>
</tr>
</tbody>
</table>

- Last element has largest element!
- Reset and compare pairs with shorter list!

What next?
Bubble Sort

Compare $j-1$ and $j$; Swap if $L[j-1] > L[j]$

Last element has largest element!

What next?
Bubble Sort

```
0  1  2  3  4
3  4  0  10  8
```

```
Comparison:

j - 1  j
1  2
```

Compare j-1 and j; Swap if L[j-1] > L[j]

What next?
Bubble Sort

```
0 1 2 3 4 5
```

```
3 0 4 10 8 11
```

```
j - 1
1
```

```
j
2
```

Compare j-1 and j; Swap if L[j-1] > L[j]

What next?
Bubble Sort

Compare \( j-1 \) and \( j \); Swap if \( L[j-1] > L[j] \)

What next?
Bubble Sort

```
0  1  2  3  4  5
3  0  4 10  8 11
```

`len = 5`

Compare j-1 and j; Swap if $L[j-1] > L[j]$
Bubble Sort

Compare j-1 and j; Swap if L[j-1] > L[j]

What next?
Bubble Sort

Reset and check pairs with shorter list

What next?
Bubble Sort

Compare $j-1$ and $j$; Swap if $L[j-1] > L[j]$

What next?
Bubble Sort

len = 4

Compare j-1 and j; Swap if L[j-1] > L[j]

What next?
Bubble Sort

0 1 2 3 4 5

What next?

Compare j-1 and j; Swap if L[j-1] > L[j]

j - 1
2

j
3

len = 4
bubble sort

0 1 2 3 4 5

j - 1

0 1

reset; compare j-1 and j; swap if L[j-1] > L[j]

what next?
Bubble Sort

Reset; Compare j-1 and j; Swap if L[j-1] > L[j]

What next?
Bubble Sort

What next?

Reset; Compare j-1 and j; Swap if L[j-1] > L[j]
Bubble Sort

Idea: bubble highest values to the end of the list; Check a shorter sublist each time

```
bubbleSort(L):
    for len in range(len(L), 1, -1):
        for j in range(1, len): # bubble
            if L[j-1] > L[j]:
                swap(j-1, j, L)
```
Bubble sort

swap(i, j, L):

1. temp = L[i]  # step 1
2. L[i] = L[j]  # step 2
3. L[j] = temp  # step 3
import swap
import random
import isSorted

def bubbleSort(L):
    """
    Sort the list L in place using bubble sort
    Param L (list): the list to sort
    Return: None
    """
    for end in range(len(L), 1, -1):
        for j in range(1, end):
            if L[j-1] > L[j]:
                swap.swap(j-1,j,L)
                print("swap", j-1, j)

if __name__ == '__main__':
    L = [10,4,3,0,11,8]
    print("Before:", L)
bubbleSort(L)
    print("After:", L, "IsSorted?", isSorted.isSorted(L))
Selection sort and Bubble sort are $O(N^2)$