

University of Toronto - Computer Science - Image

Insertion Sort

Let's start with an example:

Example

Pass 1

```
i
3 7 2 5
```

The sorted part of the list is initially empty, so we increment `i`, and move on to the next item.

Pass 2

```
       ‖i
   3   7  2  5
sorted ‖ unsorted
```

For this pass, the item at index `i` is compared with the item at index `i - 1`. Since `7` is greater than `3`, the items are in the correct order and this pass is complete.

Pass 3

```
       ‖i
  3  7  2  5
sorted ‖ unsorted
```

For this pass, the item at `i`, which is `2`, needs to be inserted into the sorted part of the list.

```
       ‖i
          5
sorted ‖ unsorted
```

We know that the value `5` will stay in its current location as we only work with the value at index `i` and the sorted part of the list.

```
       ‖i
        7  5
sorted ‖ unsorted
```

Since the `2` is less than `7`, we move the `7` over one position to the right.

```
       ‖i
     3  7  5
sorted ‖ unsorted
```

Since the `2` is less than `3`, we move the `3` over one position to the right.

```
       "
```

```
        ‖i
   2  3  7   5
sorted ‖ unsorted
```

Now, the 2 is inserted at index 0.

Pass 4

```
        ‖i
2  3  7  5
sorted ‖ unsorted
```

For this pass, the item at i, which is 5, needs to be inserted into the sorted part of the list.

```
        ‖i
2  3    7
sorted ‖ unsorted
```

Since the 5 is less than 7, we move the 7 over one position to the right.

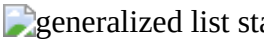
```
        ‖i
2  3  5  7
sorted ‖ unsorted
```

Since 5 is greater than 3, the 5 is inserted after the 3.

```
2 3 5 7
```

Our list is now sorted.

Generalized List States



Implementation

```
def insert(L, i):
    """ (list, int) -> NoneType

    Precondition: L[:i] is sorted from smallest to largest.

    Move L[i] to where it belongs in L[:i + 1].

    >>> L = [7, 3, 5, 2]
    >>> insert(L, 1)
    >>> L
    [3, 7, 5, 2]
    """

    # The value to be inserted into the sorted part of the list.
    value = L[i]

    # Find the index, j, where the value belongs.
```

```
    # Make room for the value by shifting.
    j = i
    while j != 0 and L[j - 1] > value:
        # Shift L[j - 1] one position to the right to L[j].
        L[j] = L[j - 1]
        j = j - 1


    # Put the value where it belongs.
    L[j] = value


def insertion_sort(L):
    """ (list) -> NoneType

    Sort the items of L from smallest to largest.

    >>> L = [7, 3, 5, 2]
    >>> insertion_sort(L)
    >>> L
    [2, 3, 5, 7]
    """

    for i in range(len(L)):
        insert(L, i)


if __name__ == '__main__':
    import doctest
    doctest.testmod()
```

Jennifer Campbell • Paul Gries
University of Toronto
