Selection Sort

Let's start with an example:

Example

Pass 1

i 3725

Variable i initially refers to 0. We need to find the smallest value in the list and swap it with item at index i. The smallest value is 2, so it is swapped with the item at i. Increase i by one.

2735

Pass 2

Note that i now refers to the index of the first item in the unsorted part of the list.

We need to find the smallest value in the unsorted part of the list and swap it with the item at index i. The smallest value is 3, so it is swapped with the item at i. Increase i by one.

Pass 3

We need to find the smallest value in the unsorted part of the list and swap it with the item at index i. The smallest value is 5, so it is swapped with the item at i. Increase i by one.

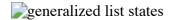
Pass 4

30/10/2024, 11:00 Selection Sort

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

There is only one item left in the unsorted part, therefore the list is sorted.

Generalized List States



Implementation

```
def get_index_of_smallest(L, i):
    """<sup>-</sup>(list, int) -> int
    Return the index of the smallest item in L[i:].
    >>> get_index_of_smallest([2, 7, 3, 5], 1)
    .....
    # The index of the smallest item so far.
    index_of_smallest = i
    for j in range(i + 1, len(L)):
        if L[j] < L[index of smallest]:</pre>
            index_of_smallest = j
    return index_of_smallest
def selection_sort(L):
    """ (list) -> NoneType
    Sort the items of L from smallest to largest.
    >>> L = [3, 7, 2, 5]
    >>> selection_sort(L)
    >>> L
    [2, 3, 5, 7]
    for i in range(len(L)):
        # Find the index of the smallest item in L[i:] and swap that
        # item with the item at index i.
        index_of_smallest = get_index_of_smallest(L, i)
        L[index_of_smallest], L[i] = L[i], L[index_of_smallest]
if __name__ == '__main__':
    import doctest
    doctest.testmod()
```

30/10/2024, 11:00 Selection Sort