Daily Coding Problem Blog

Daily Coding Problem #147

Problem

Given a list, sort it using this method: `reverse(lst, i, j)`, which sorts `lst` from `i` to j`.

Solution

This type of sorting is called pancake sorting and can be solved in a similar way as selection sort.

We iteratively put the maximum element to the end of the list using this strategy:

- First, let `size` be the size of the list that we're concerned with sorting at the moment.
- Then, we can find the position where the maximum element is in `list[:size + 1]`, say `max_ind`.
- Then, reverse the sublist from 0 to `max_ind` to put the element at the front.
- Then, reverse the sublist from 0 to `size` to put the max element to the end.
- Decrement `size` and repeat, until `size` is 0.

```
def pancake_sort(lst):
    for size in reversed(range(len(lst))):
        max_ind = max_pos(lst[:size + 1])
        reverse(lst, 0, max_ind)
        reverse(lst, 0, size)
    return lst

def max_pos(lst):
    return lst.index(max(lst))

def reverse(lst, i, j):
    while i < j:
        lst[i], lst[j] = lst[j], lst[i]
        i += 1
        j -= 1
```

This takes O(n^2) time and O(1) space.