Problem: Sorting a list in ascending order with only right shift

Algorithm: Hill Climbing (Steepest ascent)

#Initialize():

initialize a list -> [7, 1, 9, 0, 5, 8, 4, 2, 10, 0, 20]

#calculate_cost(*state*):

Counting Inversion Problem

for each element of the list:

look forward in the list and see how many elements are smaller than this element i.e. how many are in wrong order

Add up the number of disorders and return

#State_generation(*current_state*, *current_state_cost*):

min_next_cost = *INF*

min_next_state = None

for each element in the list:

next_state = swap with the forward elements of the list with this element one by one and generate one state for each swap using a for loop.

next_state_cost = **calculate_cost**(next_state)

update min_next_state & min_next_cost if applicable

take that state which has the smallest cost

if min_next_cost is smaller than current_state_cost:

return min_next_state, min_next_cost

else :

return current_state, None

#goal_test(*state*):

```
if calculate_cost(state) == 0:
        return True
else:
        return False
```

#main():

```
state = Initialize()
while(goal_test(state) is not True):
        state, cost = State_generation(state, cost)
        # When you are stuck, your cost will be None
        if cost is None:
                break
        print(state, cost)
print(state, cost)
FINISH
```