In [ ]:
```
def bubble_sort(lst):
    for i in range(len(lst)-1, 0, -1):
        for j in range(i):
            if lst[j] > lst[j+1]:
                t = lst[j]
                lst[j] = lst[j+1]
                lst[j+1] = t
#
a = [3, 5, 1, 2, 4, 6, 7]
bubble_sort(a)
print(a)                    # [1, 2, 3, 4, 5, 6, 7]
'''
p1:
    3,5,1,2,4,6,7
    3,5,1,2,4,6,7
    3,1,5,2,4,6,7
    3,1,2,5,4,6,7
    3,1,2,4,5,6,7
    3,1,2,4,5,6,7
    3,1,2,4,5,6,7
p2:
    1,3,2,4,5,6,7
    1,2,3,4,5,6,7
    ...
'''
```

In [ ]:
```
def insertion_sort(lst):
    for i in range(1, len(lst)):
        cv = lst[i]
        p = i
        while p>0 and lst[p-1] > cv:
            lst[p] = lst[p-1]
            p = p-1
        lst[p] = cv

a = [3, 5, 1, 2, 4, 6, 7]
insertion_sort(a)
print(a)
```

In [ ]:
```
def  selection_sort(lst):
    for i in range(len(lst)-1 , 0 , -1):
        p = 0
        for j in range(1, i+1):
            if lst[j] > lst[p]:
                p = j
        t = lst[i]
        lst[i] = lst[p]
        lst[p] = t

a = [3, 5, 1, 2, 7, 6, 4]
selection_sort(a)
print(a)
```

In [ ]:

```
def merge(left, right):
    i = 0
    j = 0
    a = []
    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            a.append(left[i])
            i += 1
        else:
            a.append(right[j])
            j += 1

    a += left[i:]
    a += right[j:]
    return a


def merge_sort(lst):
    if len(lst) <= 1:
        return lst
    mid = len(lst) // 2
    left  = merge_sort(lst[:mid])
    right = merge_sort(lst[mid:])
    return merge(left, right)

a = [3, 5, 1, 2, 6, 4, 7]
print(merge_sort(a))
```

In [ ]:

```
def quick_sort(lst, first, last):
   if first < last:
       p = partition(lst, first, last)
       quick_sort(lst, first, p-1)
       quick_sort(lst, p+1, last)


def partition(lst, first, last):
   p = lst[first]
   left = first+1
   right = last
   done = False
   while not done:
       while left <= right and lst[left] <= p:
           left = left + 1

       while lst[right] >= p and right >= left:
           right = right -1

       if right < left:
           done = True
       else:
           t = lst[left]
           lst[left] = lst[right]
           lst[right] = t

   t = lst[first]
   lst[first] = lst[right]
   lst[right] = t


   return right

a = [3, 5, 1, 2, 4, 6, 7]
quick_sort(a,0,len(a)-1)
print(a)
```

دانشگاه شهید مدنی آذربایجان
برنامه نویسی مقدماتی با پایتون
امین گلزاری اسکوئی
۱۴۰۰-۱۴۰۱

Codes and Projects (click here) (https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Basic-2021) slides and videos (click here) (https://drive.google.com/drive/folders/1ZsQjBJJ4UAAp9zrGxm3c4qrhnvGBUYHw)