Question 1

Name 5 sorting algorithms, also write their time complexities(best, average, worst).

Algorithm	Time Complexity		
	Best	Average	Worst
Selection Sort	Ω(n^2)	θ(n^2)	O(n^2)
Bubble Sort	Ω(n)	θ(n^2)	O(n^2)
Insertion Sort	Ω(n)	θ(n^2)	O(n^2)
Heap Sort	Ω(n log(n))	θ(n log(n))	O(n log(n))
Quick Sort	Ω(n log(n))	θ(n log(n))	O(n^2)

Question 2

Implement selection sort algorithm using Python.

In [1]:
```
def selection_sort(L):

    for i in range(len(L)-1):
        min_index = i
        for j in range(i+1, len(L)-1):
            if L[j] < L[min_index]:
                min_index = j

        L[i], L[min_index] = L[min_index], L[i]

L = [3, 1, 41, 59, 26, 53, 59]
print(L)
selection_sort(L)

print(L)
```

```
[3, 1, 41, 59, 26, 53, 59]
[1, 3, 26, 41, 53, 59, 59]
```

Question 3

Implement pop operation of the stack

In [2]:
```
stack = []

stack.append('11')
stack.append('22')
stack.append('33')

print('Initial stack')
print(stack)

print('\nElements poped from stack:')
print(stack.pop())

print('\nStack after elements are poped:')
print(stack)
```

```
Initial stack
['11', '22', '33']

Elements poped from stack:
33

Stack after elements are poped:
['11', '22']
```

Question 4

Implement dequeue operation of the queue

In [3]:
```
queue = []

queue.append('a')
queue.append('b')
queue.append('c')

print("Initial queue")
print(queue)

print("\nElements dequeued from queue")
print(queue.pop(0))
print(queue.pop(0))

print("\nQueue after removing elements")
print(queue)
```

```
Initial queue
['a', 'b', 'c']

Elements dequeued from queue
a
b

Queue after removing elements
['c']
```