

Assignment 5

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

Ans:

	<u>Best</u>	<u>Average</u>	<u>Worst</u>
<u>Selection Sort</u>	$\Omega(n^2)$	$\theta(n^2)$	$O(n^2)$
<u>Bubble Sort</u>	$\Omega(n)$	$\theta(n^2)$	$O(n^2)$
<u>Insertion Sort</u>	$\Omega(n)$	$\theta(n^2)$	$O(n^2)$
<u>Heap Sort</u>	$\Omega(n \log(n))$	$\theta(n \log(n))$	$O(n \log(n))$
<u>Quick Sort</u>	$\Omega(n \log(n))$	$\theta(n \log(n))$	$O(n^2)$

2.Implement selection sort algorithm using Python.

Ans:

```
for i in range(len(A)):
    min_idx = i
    for j in range(i+1, len(A)):
        if A[min_idx] > A[j]:
            min_idx = j

    # Swaping the found minimum element with the first element
    A[i], A[min_idx] = A[min_idx], A[i]
    print ("Sorted array")

for i in range(len(A)):

    print("%d" %A[i])
```

3.Implement pop operation of the stack.

Ans:

```
def createStack():
    stack = []
    return stack

def isEmpty(stack):
    return len(stack) == 0

def push(stack, item):
    stack.append(item)
    print(item + " pushed to stack ")

def pop(stack):
    if (isEmpty(stack)):
        return str(-maxsize -1) # return minus infinite

    return stack.pop()

def peek(stack):
    if (isEmpty(stack)):
        return str(-maxsize -1) # return minus infinite
    return stack[len(stack) - 1]

# Driver program to test above functions
stack = createStack()
push(stack, str(10))
push(stack, str(20))
push(stack, str(30))
print(pop(stack) + " popped from stack")
```