HW 4 - Cougar Bellinger

Selection Sort

```
INPUT: A array of N integers
OUTPUT: B sorted array of N integers

B = A

for i = 0 to N-1:
    min_idx = i
    for j = i+1 to N-1:
        if B[j] < B[min_idx]:
            min_idx = j

    if min_idx != i:
        temp = B[i]
        B[i] = B[min_idx]
        B[min_idx] = temp

return B
```

Input Array:

[3, 0, 4, 1, 5, 2, 9, 7, 8, 6]

Output Array:

[0, 3, 4, 1, 5, 2, 9, 7, 8, 6]
[0, 1, 4, 3, 5, 2, 9, 7, 8, 6]
[0, 1, 2, 3, 5, 4, 9, 7, 8, 6]
[0, 1, 2, 3, 4, 5, 9, 7, 8, 6]
[0, 1, 2, 3, 4, 5, 9, 7, 8, 6]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Insertion Sort

```
INPUT: A array of N integers
OUTPUT: B sorted array of N integers

B = A

for i = 1 to N-1:
    key = B[i]
    j = i - 1
    while j >= 0 and B[j] > key:
        B[j + 1] = B[j]
        j = j - 1
    B[j + 1] = key

return B
```

Input Array:

[3, 0, 4, 1, 5, 2, 9, 7, 8, 6]

Output Array:

[0, 3, 4, 1, 5, 2, 9, 7, 8, 6]
[0, 1, 3, 4, 5, 2, 9, 7, 8, 6]
[0, 1, 2, 3, 4, 5, 9, 7, 8, 6]
[0, 1, 2, 3, 4, 5, 7, 9, 8, 6]
[0, 1, 2, 3, 4, 5, 7, 8, 9, 6]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Bubble Sort

```
INPUT: A array of N integers
OUTPUT: B sorted array of N integers

B = A

for i = 0 to N-1:
    for j = 0 to N-i-2:
        if B[j] > B[j + 1]:
            temp = B[j]
            B[j] = B[j + 1]
            B[j + 1] = temp

return B
```

Input Array:

[3, 0, 4, 1, 5, 2, 9, 7, 8, 6]

Output Array:

[0, 3, 1, 4, 2, 5, 7, 8, 6, 9]
[0, 1, 3, 2, 4, 5, 7, 6, 8, 9]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Merge Sort

```
INPUT: A array of N integers
OUTPUT: B sorted array of N integers

function mergeSort(A, low, high):
    if low < high:
        mid = (low + high) / 2
        mergeSort(A, low, mid)
        mergeSort(A, mid + 1, high)
        merge(A, low, mid, high)

function merge(A, low, mid, high):
    n1 = mid - low + 1
    n2 = high - mid

    L = new array of size n1
    R = new array of size n2

    for i = 0 to n1-1:
        L[i] = A[low + i]

    for j = 0 to n2-1:
        R[j] = A[mid + 1 + j]

    i = 0
    j = 0
    k = low

    while i < n1 and j < n2:
        if L[i] <= R[j]:
            A[k] = L[i]
            i = i + 1
        else:
            A[k] = R[j]
            j = j + 1
        k = k + 1

    while i < n1:
        A[k] = L[i]
        i = i + 1
        k = k + 1

    while j < n2:
        A[k] = R[j]
        j = j + 1
        k = k + 1
```

Input Array:

[3, 0, 4, 1, 5, 2, 9, 7, 8, 6]

Output Array:

[3, 0, 4, 1] [5, 2, 9, 7, 8, 6]
[3, 0] [4, 1] [5, 2] [9, 7, 8, 6]
[0, 3] [1, 4] [2, 5] [7, 9] [6, 8]
[0, 1, 3, 4] [2, 5, 7, 9] [6, 8]
[0, 1, 2, 3, 4, 5, 7, 9] [6, 8]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]