CMSC 11: Sorting Algorithms
S-4L and GH-4L

```
procedure bubbleSort( A[10] ) defined as:
  do
    swapped := false
    for each i=0 to 10 - 2 do:
      if A[ i ] > A[ i + 1 ] then
        swap( A[ i ], A[ i + 1 ] )
        swapped := true
      end if
    end for
  while swapped==true
end procedure
```

```
function merge_sort(m)
    var list left, right, result
    if length(m) ≤ 1
        return m

    var middle = length(m) / 2
    for each x in m up to middle
         add x to left
    for each x in m after middle
         add x to right
    left = merge_sort(left)
    right = merge_sort(right)
    result = merge(left, right)
    return result
```

```
insertionSort(array A)
begin
    for i := 1 to length[A]-1 do
    begin
        value := A[i];
        j := i-1;
        while j ≥ 0 and A[j] > value do
        begin
            A[j + 1] := A[j];
            j := j-1;
        end;
        A[j+1] := value;
    end;
end;
```

```
function selection_sort(array A[n]){
for i = 0 to n-2 do{
    min = i
    for j = (i + 1) to n-1 do{
        if A[j] < A[min]
            min = j
    }
    swap A[i] and A[min]
}
}
```