Andrew Sivaprakasam | Warm-Up 1 Write-up

GitHub Repo: `https://github.com/sivaprakasaman/Numerical_Methods_BME/`

1 Divide-and-Conquer for Integer Multiplication

2 Sorting Algorithms

Note: My typset algorithms are *zero-based*– they assume indexing starts at zero, not the case in Matlab.

Bubble Sort: Typset Algorithm. Worst/Average Case = $O(n^2)$, Best Case = $O(n)$

```
function bubbleSort($a[1, ..., k]$)
input  : An unsorted array, $a$
output: The sorted array
Iterate through each element i in a, swapping it with the element prior if element i-1 is
  greater than element i
swapped = true;
while swapped do
    swapped = false;
    for $i \leftarrow 1$ to $k$-1 do
        if $a[i-1] > a[i]$ then
            swap: $a[i-1] \leftrightarrow a[i]$;
            swapped = true;
        end
    end
end
```

Insertion Sort: Typset Algorithm. Worst/Average Case = $O(n^2)$, Best Case = $O(n)$

```
function insertionSort($a[1, ..., k]$)
input  : An unsorted array, $a$
output: The sorted array
Incrementally traverse through each element i in a, sliding a[i] left through the sorted array,
  until it is ranked appropriately
$i = 1$;
while $i < k$ do
    $j = i$;
    while $j > 0$ and $a[j-1] > a[j]$ do
        swap: $a[j] \leftrightarrow a[j-1]$;
        $j \leftarrow$ j - 1;
    end
    $i \leftarrow i + 1$;
end
```

Selection Sort: Typset Algorithm. Worst/Average/Best Case = $O(n^2)$

```
function selectionSort($a[1, ..., k]$)
input  : An unsorted array, $a$
output: The sorted array

for $i \leftarrow 0$ to $k\text{-}1$ do
    starting with element i, compare i against elements ahead
    tempMin = $i$;
    for $j \leftarrow i + 1$ to $k$ do
        if $a[j] < a[tempMin]$ then
            tempMin = $j$;
        end
    end
    if tempMin does not equal i then
        swap: $a[i] \leftrightarrow a[\text{tempMin}]$;
    end
end
```