

CSSE 230
Homework assignment

For each of the algorithms, determine the big-Oh worst-case runtime. Please show your work.

(a) Maximum Contiguous Subsequence, version 1

```
public static int maxSubSum1(int[] a){
    int maxSum = 0;
    for(int i = 0; i < a.length; i++)
        for(int j = i; j < a.length; j++) {
            int thisSum = 0;
            for(int k = i; k <= j; k++)
                thisSum += a[k];
            if(thisSum > maxSum) {
                maxSum = thisSum;
                seqStart = i;
                seqEnd = j;
            }
        }
    return maxSum;
}
```

(b) Bubble Sort

ALGORITHM *BubbleSort*($A[0..n-1]$)

//Sorts a given array by bubble sort

//Input: An array $A[0..n-1]$ of orderable elements

//Output: Array $A[0..n-1]$ sorted in nondecreasing order

for $i \leftarrow 0$ **to** $n-2$ **do**

for $j \leftarrow 0$ **to** $n-2-i$ **do**

if $A[j+1] < A[j]$ swap $A[j]$ and $A[j+1]$

(c) Brute-Force String Matching

```
ALGORITHM BruteForceStringMatch( $T[0..n - 1]$ ,  $P[0..m - 1]$ )
//Implements brute-force string matching
//Input: An array  $T[0..n - 1]$  of  $n$  characters representing a text and
//       an array  $P[0..m - 1]$  of  $m$  characters representing a pattern
//Output: The index of the first character in the text that starts a
//        matching substring or  $-1$  if the search is unsuccessful
for  $i \leftarrow 0$  to  $n - m$  do
     $j \leftarrow 0$ 
    while  $j < m$  and  $P[j] = T[i + j]$  do
         $j \leftarrow j + 1$ 
    if  $j = m$  return  $i$ 
return  $-1$ 
```

(d) Maximum contiguous subsequence, version 2

```
public static int maxSubSum2(int[] a){
    int maxSum = 0;
    for(int i = 0; i < a.length; i++){
        int thisSum = 0;
        for(int j = i; j < a.length; j++){
            thisSum += a[j];
            if(thisSum > maxSum){
                maxSum = thisSum;
                seqStart = i;
                seqEnd = j;
            }
        }
    }
    return maxSum;
}
```

(e) Maximum contiguous subsequence, version 3

```
public static int maxSubSum3(int[] a){
    int maxSum = 0;
    int thisSum = 0;
    for(int i = 0, j = 0; j < a.length; j++){
        thisSum += a[j];
        if(thisSum > maxSum){
            maxSum = thisSum;
            seqStart = i;
            seqEnd = j;
        } else if(thisSum < 0) {
            i = j + 1;
            thisSum = 0;
        }
    }
    return maxSum;
}
```