CS225 Homework 9 – Aaron Van De Brook

Problem 1:

in: a[n]

out: b[n]

for i=0 to n do

count = 0

for j=0 to n do

if a[i] > a[j] then

count += 1

fi

od

b[i] = count

od

Worst case complexity: O(n2) – There is a loop that calls a loop that makes n comparisons n number of times, therefore the worst-case complexity of this algorithm is O(n2).

Problem 2:

in: a[n][n]

out: max

count[n] = {0}

for i=0 to n do

count[i] = 0

for j=0 to n do

if a[i][j] == 1 then

count[i]++

fi

od

od

max = count[0]

for x=1 to n do

if count[x] > count[x-1] then

max = count[x]

fi

od

Worst case complexity: O(n2) – There are 2 loops that make n comparisons n times making the complexity O(n) + O(n), one of the loops calls another loop that makes n comparisons n times making the final complexity O(n) \* O(n) + O(n) or O(n2) + O(n) and taking the highest degree of the final complexity we get O(n2) for the worst case complexity for this algorithm.