1

Q1 (4 points)

1 page submitted

Count the number of assignments and comparisons (all of them) for each of the following algorithms using summation notation.

a) **Algorithm** Loop1(n):
   $p \leftarrow 1$
   for $i \leftarrow 1$ **to** $n^2 + 1$ **do**
      $p \leftarrow p \cdot i$

b) **Algorithm** Loop2(n):
   $s \leftarrow 1$
   for $i \leftarrow 0$ **to** $n^2 + 1$ **do**
      for $j \leftarrow 1$ **to** i **do**
         $s \leftarrow s + i$

2

Q2 (4 points)

1 page submitted

Describe a recursive algorithm for finding both the minimum and maximum elements in an array A of n elements. Your method should return a pair (a, b) where a is the minimum element and b is the maximum element. Count the assignments (including returns) and comparisons in order to derive a recurrence equation for the worst-case runtime of your algorithm.

3

Q3 (4 points)

1 page submitted

Consider the following recurrence equation, defining a function T(n):

$T(0) = 1$ and $T(n) = T(n-1) + 2^n$ when $n \geq 1$.

Use top-down repeated substitution to derive the closed form of this equation.

4

Q4 (4 points)

1 page submitted

Consider the following recurrence equation, defining a function T(n):

$T(1) = 1$ and $T(n) = T(n-1) + n$ when $n \geq 2$.

Prove by induction that $T(n) = \frac{n(n+1)}{2}$.

5

Q5 (4 points)

1 page submitted

Consider the following algorithm which adds up the elements of an array. Use a loop invariant argument to prove that arraySum() is correct.

```
Algorithm arraySum(A,n)
   sum ← A[0]
   for k ← 1 to n-1 do
      sum ← sum + A[k]
   return sum
end
```