CS-203 – MIDTERM EXAM – 2022 SUMMER

GIUSEPPE TURINI – KETTERING UNIVERSITY

Instructions

- This exam is take-home, open-book, open-notes, and individual (no collaboration).
- Each part indicates the points awarded if correctly answered (partial credit available).
- Submit your solution as a single PDF file, via email using your Kettering account.
- The submission deadline is: Sunday 31 July 2022, before the end of the day.

Student Information

- Student full name (readable) and signature:

Exercise 1 (50 points)

Consider the following *iterative* (*i.e.*, *non-recursive*) algorithm:

```
void mysteryAlgorithm1(int[] A) {
   int n = A.length;
   boolean swapped = false;
   do {
      swapped = false;
      for(int i = 0; i < n-1; i++) {
         if(A[i] > A[i+1]) {
            int swap = A[i];
            A[i] = A[i+1];
            A[i+1] = swap;
            swapped = true;
         }
      }
      for(int i = n-2; i >= 0; i--) {
         if(A[i] > A[i+1]) {
            int swap = A[i];
            A[i] = A[i+1];
            A[i+1] = swap;
            swapped = true;
         }
      }
   } while(swapped);
}
```

Analyze this algorithm (*i.e.*, mysteryAlgorithm1), and:

a Determine what this algorithm computes, briefly explaining its strategy (10 points).

b Determine input size and basic operation, briefly explaining your choices (10 points).

c Express the basic operation count as a summation (10 points).

d Convert the basic operation count summation into a closed-form expression (10 points).

e Find the efficiency class of the basic operation count, using proper notation (10 points).

Note If necessary, perform this analysis for both the best-case and worst-case.

Exercise 2 (50 points)

Consider the following *recursive* algorithm:

```
int mysteryAlgorithm2(int n) {
   if(n == 0) {
      return 1;
   }
   else {
      int tmpRes = mysteryAlgorithm2(n-1);
      int res = n;
      for(int i = 1; i < tmpRes; i++) {
         res += n;
      }
      return res;
   }
}
```

Analyze this algorithm (*i.e.*, mysteryAlgorithm2), and:

a Determine what this algorithm computes, briefly explaining its strategy (10 points).

b Determine input size and basic operation, briefly explaining your choices (10 points).

c Write the recursive definition of the algorithm computation (5 points)

d Write the recursive definition of basic operation count (5 points)

e Convert the basic operation count into a closed-form expression (10 points)

f Find the efficiency class of the basic operation count, using proper notation (10 points).

Note If necessary, perform this analysis for both the best-case and worst-case.