COSC2010/COSC2100: HW2

Due: 1:59PM Sept. 10

Turn in a softcopy to turnin before due time

Total: 20pts

Your work should be typed and saved in a text file named HW2.txt.

1. Follow the pseudo code convention discussed in class, write an algorithm for each problem below. Indent your pseudo codes correctly!
    a. (2pts) Design an algorithm that takes an array, namely *theArray*, of *n* positive numbers, and determines the maximum value of A[j] + A[i], for j ≠ i. The algorithm returns the maximum value.

    b. (2pts) Consider three arrays that are populated with integers, e.g., positive, negative, or zero. Design an algorithm that takes three arrays, namely arrayA, arrayB, arrayC, each with size *n*, and checks if there exists a triple which adds up to zero. The three elements in a triple must come from different arrays. The algorithm returns true if such a triple exists, and false otherwise.

    c. (2pts) Consider sorting *n* integer numbers stored in array *theArray* with the following algorithm. The algorithm first finds the smallest element of *theArray* and exchanges it with the element indexed by 0, and then finds the second smallest element of *theArray* and exchanges it with the element indexed by 1. The algorithms continues in this matter for n − 1 times before it finishes. The algorithm takes *theArray* and *n* as inputs, and doesn't return a value.

    d. (4pts)The Hailstone Series is generated using the algorithm described by the following plain English:
        i. Pick a positive number (0 or greater), and place this number in the beginning of the series.
        ii. If it is odd: triple the number and add one; append the resulting number at the end of the series.
        iii. If it is even: divide the number by two; append the resulting number at the end of the series.
        iv. Go back to step 2.

        This series will eventually reach the repeating "ground" state: 4, 2, 1, 4, 2, 1
        Here is the sequence generated for an initial value of 6: 6, 3, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1

        Write the algorithm in pseudo code with the following details:
        - Convert the high-level description of the algorithm above into pseudo-code.
        - Add statements in your pseudo code to keep track of the following values and display them during execution:
            - The number of items that are currently in the sequence.
            - The largest value that is stored in the sequence.

- Assume the "ground" state commences when the integer 4 is computed. At this stage terminate computation of the series.
- Your algorithm should terminate without calculating the series if 0 is input as the positive number.

2. We have analyzed in class that the `bubblesort` algorithm is $O(n^2)$.
    a. (5pts) Suppose it takes 1 second to run the `bubblesort` algorithm on an array of 10,000 elements on a computer A. How many seconds does it take to run `bubblesort` on an array of 40,000 elements?
    b. (5pts) Suppose computer B is 10 times faster than computer A. What array size can be sorted by `bubblesort` on computer B in 10 seconds?