

Tuesday, February 18, 2020

CSI-345 Advanced Data Structures

Midterm Project

In all the questions you cannot use any predefined method from a .Net library or Java library  
example if you need to have the max value, cannot use array.Max(), you have to create/define your own max method.

Array:

1. Given an array of ints, write a method that returns the second pair of entries that add up to zero

You can create a class Pair(int x,int y) to hold 2 values, x and y that add up to 0

Or you can just return a string : $”[{x},{y}]” that contains the two values

1. Given an unsorted array of ints, write a method that reverses the array in place, that is without any temp array, queue or stack used.
2. Given two arrays array1 and array2 both of type int. write a method that takes 2 arrays and return an array that contains the common elements. This method is usually called Intersection

If array1: {-5, 6, 3, -2, 10, -12, 20, 11} and array2: {7, 20, 9, -5, 2, 14,3} then the intersection array that is returned should contain: {20,-5,3}. The order does not matter.

1. Write a method that takes a sorted array of ints and removes all the duplicates

If array is: {7, 10,10, 12, 15, 17, 17, 17, 17, 20, 25, 33, 40,40,40, 45, 45, 49}

It will end up to be: {7, 10,12, 15, 17, 20, 25, 33, 40, 45, 49}

Use the linkedList we did in class (it is on canvas Module/WEEK-4)

1. Write a method (public void reverse()) that reverses the linkedlist. No additional temp array, queue, stack or linkedlist can be used.
2. Write a sort method (using either bubblesort or insertionsort) to sort the linkedlist
3. Write a method that removes duplicates from a sorted linkedlist

Use .net Stack<int> or java Dequeue<int> class

1. Write a method that takes a stack of ints and removes the last entry from it (keep the rest in their original order
2. Write a method that takes a stack<int> and moves the first half of the stack and swap it with the top half of the stack

If a stack has:

|  |
| --- |
| 7 |
| 3 |
| 15 |
| 4 |
| 9 |
| 20 |
| 33 |
| 25 |
| 31 |
| 27 |

After the swap method, the stack would have:

|  |
| --- |
| 20 |
| 33 |
| 25 |
| 31 |
| 27 |
| 7 |
| 3 |
| 15 |
| 4 |
| 9 |

You will be graded on whether your algorithm works as well as whether it more efficient, that is done with the least amount of code (or loops)

Use Queue<int> for .net or Dequeu<int> for java

1. Write a method that takes a Queue of ints and position the max entry at the front of the queue. If queue: 10🡪2🡪15🡪7🡪11 then you code is to move the max 15 to the front to have 15🡪10🡪2🡪7🡪11