1.

CREATE an array a[][] from the arrays

FUNCTION mergeSortedArrays(given the a[][] and n)

WHILE LOOP **index goes through loop until index value is past end of array**

         WHILE LOOP **index goes through loop until index value is past end of current array**

Add array element to the tree using heap.add(key,value) to far left location of the lowest level

 // Check if inserted node is in right location. If not, move it to right location.

  WHILE LOOP inserted node has a parent

          IF inserted node < parent

             swap inserted node with parent

        ELSE

    break (because inserted node is in correct position)

          END IF

               END WHILE LOOP

         END WHILE

    END WHILE

CREATE new array B to store the n smallest values

SET B[0] to heap.find-min()

      SET counter to 1

      WHILE LOOPcounter < n

              TRAVERSE the tree to get the next smallest value

Add value to B array

INCREMENT counter by 1

      END WHILE

RETURN Array B

END FUNCTION

2.

FUNCTION findLongestSerial(takes in array)

SET A[index] to start of array

SET count to 1

SET highestCount to 1

**WHILELOOP index goes through loop until index value is past end of array**

**SET initial to A[index] (initial set to current slot it has looped on, starting at beginning)**

**FORLOOP where initial + 1 to start, runs until the end of array, and increments index by 1**

**IF initial = A[index] - 1**

**ADD count by 1**

**SET A[index] as the new initial**

**END IF**

**END FORLOOP**

**IF count > highestCount**

**SET highestCount to count value**

**END IF**

**SET count to 1**

END WHILE

RETURN highestCount

END FUNCTION

3.

FUNCTION DistinctElements(contains array as A[index], k value for sections, and array length of n)

     CREATE empty hash map called hash

CREATE array B

SET count to 0

//Establish and traverse through the first k section and store count of each element in hash

FOR LOOP where index is set to 0, ends when index passes k, and index increments by 1

        IF hash[Array[index]] is equal to 0

INCREMENT count by 1

ENDIF

INCREMENT hash[Array[index]] by 1

END FOR LOOP

INSERT count into first slot of array B

ADD index by 1 to move on to next section and go through rest of A

FOR LOOP where index set to k, ends when index passes n-k+1, and index increments by 1

Remove first element of previous k section (IF only one occurrence of the element, DECREMENT the count by 1)

  DECREMENT hash[Array[index]] by 1 to account for removed element

Add new element to current section (IF element appears for the first time, increment count by 1) with INCREMENT hash[Array[index]] by 1

INSERT count into next available slot of array B and move slot over one (we can keep count as is now since it will adjust as needed)

   END FOOR LOOP

RETURN Array B

END FUNCTION

4.

Part 1

FOR LOOP where index is set to 1, ends when index passes n, and index increments by 1

IFA[index] + index is equal to n + 1

RETURN A[index]

END IF

END FOR LOOP

RETURN statement saying “no solution”

Part 2

I don’t know