

CSE 332 Data Abstractions, Spring 2016

Homework 5

Due: **Wednesday, May 11, 2016** at the BEGINNING of lecture. Your work should be readable as well as correct.

Problem 1: Sorting Phone Numbers

The input to this problem consists of a sequence of 7-digit phone numbers written as simple integers (e.g. 5551202 represents the phone number 555-1202). The sequence is provided via an `Iterator<Integer>` - *you do not get an array containing these phone numbers and you cannot go through the iterator more than once*. No phone number appears in the input more than once but there is no limit on the size of the input. You may assume that phone numbers will not start with 0, although they may contain zeroes otherwise.

Write precise pseudocode for a method that prints out the phone numbers (as integers) in the list in ascending order. Your solution must not use more than 2MB of memory. (Note: It cannot use any other storage – hard drive, network, etc.) In your pseudocode you may only declare variables and arrays of these unsigned data types (these are not real Java data types): `bit`(1 bit), `byte`(8 bits), `short`(16 bits), `int`(32 bits), `long`(64 bits). **Explain** why your solution is under the 2MB limit.

Problem 2: hasOver

Submit the solution to this problem using Gitlab.

Use the `ForkJoin` framework to write the following method in Java:

```
Public static Boolean hasOver(int val, int[] arr, int seqCutoff)
```

Returns true if arr has any elements strictly larger than val.

For example, if arr is [21, 17, 35, 8, 17, 1] then

`hasOver(21, arr) == true` and `hasOver(35, arr) == false`.

Your code must have $O(n)$ work, $O(\log n)$ span, where n is the length of arr, and must use the `seqCutoff` argument.

Problem 3: getLongestSequence

Submit the solution to this problem using Gitlab.

Use the `ForkJoin` framework to write the following method in Java:

```
Public static Boolean getLongestSequence(int val, int[] arr, int seqCutoff)
```

Returns the length of the longest consecutive sequence of val in arr.

For example, if arr is [2, 17, 17, 8, 17, 17, 0, 17, 1] then

`getLongestSequence(17, arr) == 3` and `getLongestSequence(35, arr) == 0`.

Your code must have $O(n)$ work, $O(\log n)$ span, where n is the length of arr, and must use the `seqCutoff` argument. We have provided you with an extra class `SequenceRange`. We recommend you use this class as your return value, but it is not required.