CSE 250 WA2: PA1 and PA2 Runtime Analysis **Fall 2019**

Due: Wednesday, 10/16/2019 before 5:00PM

Total points = 40

1 Instructions

Expect this assignment to take 1-3 hours in total.

Problem 1. (2+2 points each) Recall PA1's `DataEntryStore` and the `insert` and `remove` methods. In the best case, we could `insert` at a cost of $O(1)$ per insertion and remove was $\Theta(n)$. To appreciate the benefit of the `DataEntryStore`, suppose that we changed our implementation so that `dataArray` had the type `FixedArrayList[TaxEntry]`. Then we can simply kept the oldest value stored at the lowest index (`0`) and the newest value stored at the highest index (`numStored-1`), doing away with the idea of linked nodes. Then we have a new implementation of `insert` and `remove` as follows:

```
def insert(entry: TaxEntry) = {
  if (numStored == capacity) {
    dataArray.remove(0)
    numStored -= 1
  }
  dataArray.insert(numStored,entry)
  numStored += 1
}

def remove(entry: TaxEntry) = {
  val removed = false
  for (i <- dataArray.length-1 to 0 by -1)
    if (dataArray(i) == entry) {
      dataArray.remove(i)
      removed = true
      numStored -= 1
  }
  removed
}
```

1(a) Explain a scenario that causes the best-case execution (runtime) for `insert`. Provide the runtime analysis for this situation.

1(b) Explain a scenario that causes the worst-case execution (runtime) for `insert`. Provide the runtime analysis for this situation.

1(c) Explain a scenario that causes the best-case execution (runtime) for `remove`. Provide the runtime analysis for this situation.

1(d) Explain a scenario that causes the worst-case execution (runtime) for `remove`. Provide the runtime analysis for this situation.

Problem 2. (2+2+6 points per function) Provide a full runtime analysis of the methods `GroupByStore.insert` and `GroupByStore.regroup` based on your submission for PA2. For your analysis of each, do the following:

2(a) Provide a summary of your solution code in pseudocode.

- Use the lecture notes for examples of this: e.g., `insert` and `remove` are good form to follow for summarizing as in the `ArrayList` notes.
- This should not be a copy of your code, but an English description of what happens.

2(b) Create a printout of your code containing the `insert` and `regroup` methods. Then, based on your pseudocode, box in the instructions that correspond to each step in your pseudocode you wrote in 2(a). Each idea should correspond to some part of your code. You should also include the definitions of any member variables that are utilized.

- Ideally, you should number your pseudocode steps and then number the corresponding box(es).

2(c) Analyze the runtime and additional space allocation of your code.

- Provide these as tight Big-O bounds along with your derivations.

Due to the inability to distinguish efficiency on Autolab, if your solution is inefficient your score here will have 5 points deducted for the PA2 efficiency requirement. This is here because program behavior must be checked by hand. You will also receive deductions here if your code uses ineligible structures.

Problem 3. (4 points) Suppose groups of TaxEntry objects were stored in `ArrayBuffer` objects instead of in linked lists within `groupings`. Briefly describe what change(s) might need to be made to address this.

2 Submission

Your written solution may be either handwritten and scanned or typeset. Either way, you must produce a PDF that is legible and displays reasonably on Autolab. This PDF should be submitted on Autolab to WA2 Written. You can view the PDF after you submit by clicking the view icon (magnifying glass with a +).

- What you see is what we will see when we grade it. If the submission is obnoxiously zoomed in, rescan your document.
  - If you open your PDF in Adobe (or another viewer) the Page Size should be something around 8.5in x 11in (or the corresponding dimensions of A4).
  - If you take a picture and use certain conversions, it will likely be around 30in x 40in, which is not reasonable and will not be graded.
- If you type your solutions in Microsoft Word (or some other non-pdf format), ensure you submit a PDF file and not the document you edited. **If you submit a document that does not display or is malformed, your submission will not receive credit.**

3 Late Policy

Written submissions face a 50%-penalty per day late. No grace day usage is allowed.