Name: Pheakdey Luk **Assignment 6**

ID: 986591

R-4.14 Which, if any, of the following algorithms, bubble-sort, heap-sort, insertion sort, merge-sort, and quick-sort, are stable? Briefly justify your answer.

**Answer:**

* None of these algorithms are stable.
* Justification:

Stable sorting algorithms maintain the relative order of records with equal keys (i.e. values). Therefore, a sorting algorithm is stable if whenever there are two records R and S with the same key and with R appearing before S in the original list, R will appear before S in the sorted list.

R-4.16 Is the bucket-sort algorithm in-place? Why or why not?

**Answer:**

No, bucket sort is not in-place because we need to move the items into the bucket for sorting.

C-4.13 Suppose we are given two sequences A and B of n elements, possibly containing duplicates, on which a total order relation is defined. Describe an efficient algorithm for determining if A and B contain the same set of elements (possibly in different orders). What is the running time of this method?

**Answer:**

Algorithm isSameSetElements**(**A**,**B**)**

Input**:** Sequence A**,** B

Ouput**:** **true** **if** they are elements of same set**,**othewise **false**

**if** A**.**size**()** **=** B**.**size**()** then

D1**<-** Dictionary**(**hastable**)**

D2**<-** Dictionary**(**hastable**)**

**for** each x of A **do**

cnt**<-**D1**.**findElement**(**x**)**

**if** cnt **!=** NO\_SUCH\_KEY then

D1**.**insertItem**(**x**,** cnt **+** 1**)**

**else**

D1**.**insertItem**(**x**,** 0**)**

**for** each x of B **do**

cnt**<-**D2**.**findElement**(**x**)**

**if** cnt **!=** NO\_SUCH\_KEY then

D2**.**insertItem**(**x**,** cnt **+** 1**)**

**else**

D2**.**insertItem**(**x**,** 0**)**

**for** each x of A **do**

**if** D1**.**findElement**(**x**)** **!=** D2**.**findElement**(**x**)** then

**return** **false**

**return** **true**