COEN 352 Summer 2020 Assignment 2

Assignment 2 continues the development on WarehouseInventory database. Please refer to Assignment 1 for WarehouseInventory database description. In database, the term *index* is a data structure for query and retrieving data with improved speed. For example, based on the *Inventory Values*, the inventory records can be sorted. However, sorting will result in exchange of elements at positions. Even worse, if the query changes to other field such as *QTY*, then the sort will exchange the records according the new requested field. Certainly this is not time efficient. The solution is to build an index for a specific query. An index on the field of *Inventory Values* is a data structure that only keeps the position of records according to a certain criteria, such as the descending order of the inventory values of each record. The index does not contain a record itself, but only the position.

Problem 1. (10 marks) The KVpair class should **implements** Comparable<KVpair>, and the member function **public int** compareTo(KVpair o). This compareTo function is defined to compare the value of one specific data member such as *Inventory Values* of the inventory record that the KVpair object represents.

Problem 2. (20 marks) Add one new member function to the Dictionary ADT as below. Revise your implementation of Dictionary and provide the member function body to your Dictionary class. The return is an int[] that contains the sorted position in descending order of the original records in the Dictionary. The original order of the record SHOULD NOT be changed. An illustration is shown in the Figure 1 below. The index created is the sorted position rather than swapping records to the sorted position. The solution should revise one big-Theta(n log n) sort algorithm (quick sort or merge sort) to provide the implementation.

```
public int[] createDesendingIndex();
```

Problem 3. (20 marks) In your WarehouseInventory class,

- Program a function to initialize the database as the example given from the excel sheet.
- Program a clean function to empty the database
- Program a sort function that uses the MaxHeap sort algorithm to sort the inventory records in the descending order based on the *Inventory Values*

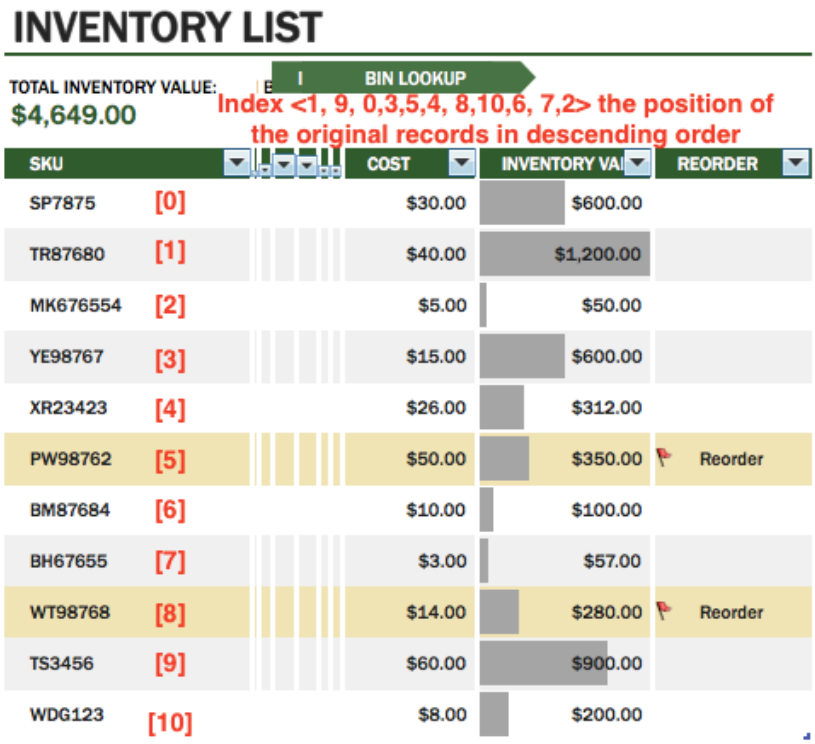

Figure 1. An illustration of index on Inventory Values in descending order

Submission Specification

1. Program the solution in a single project, thus under one ***src*** folder
2. The src should contain all the Java files
3. The src folder should be archived together as a single file, following the naming convention [SID_1]_[SID_2]_A2.zip or
   [SID_1]_[SID_2]_A2.gz or
   [SID_1]_[SID_2]_A2.tar

   No .rar file is accepted. Do not following the naming convention will cause delays in releasing the marking grade.

4. Submission is due on **July 31st 23:59**. Grace period is 6 hours later. Cut off data with 20% penalty is 24 hours later. Submission is through Moodle site ONLY. Submission in emails is not accepted.