**CECS 302 Homework Assignment 4 (Due 4/24/23)**

**1) (30 pts)** This problem deals with **sorting algorithms**.

a. Explain the difference between the terms **Maintained Sorting** and **Sorting On Demand**. Provide at least one example of an application where each one is preferable.

b. Explain the principal differences between the **Insertion Sort** and **Heap Sort** algorithms. Be certain you explain i. any requirements the algorithms have to be used, ii. a general idea of how the algorithms operate, and iii. which algorithm is the most efficient in the **worst-case scenario**. Your response should be 1-2 paragraphs in length.

**2) (30 pts)** This problem deals with **sets** and **set implementations.**

a. Explain the principal differences between the **Unsorted List ADT** and the **Set ADT**.

b. Provide an example when using the bit-vector implementation of a set can be advantageous, and another example when it is a poor choice.

**3) (40 pts)** This problem deals with **hash tables** and **hashing functions.** Your responses to each sub-part should be 2-5 sentences in length.

a. Explain the principal efficiency advantage of hash tables in comparison to other data structures we have observed thus far.

b. Explain what issues can hamper this performance advantage, and what steps are taken to mitigate the slow-down.

c. Suppose employees at a company are assigned an 8-digit employee ID according to the following criteria:  
i. The first four digits correspond to a department code (ex: the accounting department may have code 0051, the IT department code 3505, etc.)  
ii. The second four digits correspond the order in which the employee was hired (ex: the second employee would have the last four digits “0002,” the one-hundredth “0100,” etc.)  
Now suppose you are given the task to store employee information according to employee ID in a small hash table and are given the following options for the hashing function:   
i. Compute the hash index using the **first** four digits of the employee ID, mod 53.   
ii. Compute the hash index using the **last** four digits of the employee ID, mod 53.  
Assuming the company has 100+ employees and 5 departments, explain which approach is better for this dataset, and why. Should your selection change if option ii only used the last *two* digits of the ID, mod 53. Explain why or why not.

**Responses to all problems should be in .doc(x) or .pdf format. Please submit your assignment to Blackboard as a single file (not zipped, in this case) with the filename “LN\_FN\_4” where LN is your last name and FN is your first name.**

1.a The difference between maintained sorting and Sorting on Demand, the maintained sorting keeps the data in relative order of equal elements and the Sorting on Demand refers to the sorting the data only when its needed. One example for maintained sorting could be on e-commerce website where the fruits and products need to be maintained sort when the products arrived, which department that product belong etc. One example for sorting on demand where we can use in applications where the database is too large, on my CSE 335 Project we use the sorting demand to sort a column of data in ASC or DES order.

1b. The insertion and Heap sort are both sorting algorithms used to arrange elements in a particular order. Insertion sort is very simple to implement where it is used when we are using number of elements in small. It has a worst-case time complexity of O(n^2) making it inefficient because it is going to take more time for large databases. Heap sort does work by creating a heap DS from array. it requires additional memory to create the heap, the worst-case time is O(n log n) where heap is more efficient for large datasets.

2a. Unsorted List ADT and Set ADT are fundamental data structure use in computer science. They are both allow us to get access and manipulate our data. If we use Unsorted List ADT, the data can be stored in any order we can access to this data by their index position in the list. The Set ADT does not have inherent order and does not provide an index position instead we could get access through some operations like union, intersection, and difference.

2b. Each item in a bit-vector implementation is represented by a single bit in the array, where a value of 1 indicates that the item is present in the set and a value of 0 indicates that there is nothing present. Utilizing a bit-vector has the benefit of providing us with efficient storage that enables us to store numerous values of the same data type. We will need to conduct set operations like intersection, union, etc. if we utilize a bit vector that contains a set of unique strings. A bit-vector of the set might not be the best choice in this situation because there are likely many potential strings and they would not all fit in memory. Additionally, some searching algorithms use non-dynamic data structures like vectors.

3a. The advance of a hash table is efficiency when we are getting a return, when the item is indexed that going to help our code to get better response back. The hash tables are preferer in the world of programming because its speed to perform insertion, deletion and searching.

3b. if you don’t know the right unique key there is going to present a problem while you are requesting a data back. Make sure that the unique key is correct on your dataset that way the result is going to be accurate, and the answer back is going to be correct.

3c.

