**Search & Sort Algorithms Used**

In the bank management system computer program, there were different algorithms used for searching and sorting user accounts data. In the following section it is provided a description of how these algorithms were used to properly execute program functions.

**1. Linear Search**

Linear search, also known as sequential search, is a method for finding a target value within a list or array. It involves checking each element of the list one by one until a match is found or the entire list has been traversed.

In Query (Search) function, linear search was used to search for a user account number in an array of structures. Each structure contains the user’s account number that is required to search for. In the following section it is provided a step-by-step explanation of how linear search works in the function.

**1-Start from the beginning:** The search begins by setting an integer variable (flag) by zero to indicate that no account number has been found so far. Then examining the first account number in the first structure.

**2-Compare with the target account number:** Check if the current number is equal to the target number searched for.

**3-Match found:** If the current element is equal to the target, the search is complete, and the index or position of the element is known from loop index (i).

**4-No match:** If the current element is not equal to the target, move to the next structure in the array.

**5-Repeat:** Steps 2-4 are repeated until a match is found or the end of the array is reached.

**6-End of Array:** If the end of the array is reached without finding a match, the search is done but no account name is found, and the flag variable will still contain a value of zero.

**2. Bubble Sort**

**NOTE: In the program, bubble sort was used in several functions (SortByName,**  **SortByBalance, and** **SortByDate). But the same sort algorithm was used regardless of the function.**

Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. The pass through the list is repeated until the list is sorted. It is provided in the following section a step-by-step explanation of how bubble sort was used in the program to sort accounts’ structures based on several factors.

**1-Start at the Beginning:** Begin with the first structure value in the array.

**2-Compare Adjacent Elements:** Compare the current structure value with the next value.

**3-Swap if Necessary:** If the current value is greater than the following value, swap them.

**4-Move to the Next Pair:** Move to the next pair of structures (i.e., advance to the next index).

**5-Repeat Until the End:** Repeat steps 2-4 until you reach the end of the list. After the first pass, the smallest element is guaranteed to be in its final position at the end of the list.

**6-Repeat for the Unsorted Part:** Repeat steps 1-5 for the remaining unsorted elements. After each pass, the next largest element will be in its correct position.

**7-Continue Until Sorted:** Continue these steps until the entire list is sorted.

**Notes:**

* The number of passes in the worst case will be less than the number of elements by 1.
* In every pass the number of elements iterated on will be the total number of elements subtracted from the number of passes. That is because in every pass it is sure that the last item will be in the correct position.