



General Sir John Kotelawala Defence University

Department of Computer Science

CS2062 – Data Structures and Algorithm II (Intake 40)

Lab Sheet 05

1. Sort the following array by using Radix sort. Use the following algorithm and code block to implement the program. The radix in the **decimal system** is **10**.

{170, 45, 75, 90, 802, 24, 2, 66 }

Radix Sort Algorithm Steps

1. **Initialize Data Structures:**
 - Create a 2D array radixArray of size 10 x n (where n is the length of the input array) to temporarily hold the elements based on their current digit.
 - Create a 1D array counts of size 10 to count the occurrences of each digit (0-9).
2. **Find the Maximum Value:**
 - Traverse the input array to find the maximum value (maxVal). This helps determine the number of digits in the largest number.
3. **Initialize Exponent:**
 - Set the initial exponent exp to 1. This represents the current digit position (1 for units, 10 for tens, 100 for hundreds, etc.).
4. **Sort Elements by Each Digit:**
 - **Repeat the following steps until maxVal / exp is greater than 0:**
 1. **Distribute Elements:**
 - For each element in the input array:
 - Calculate the digit at the current exp position using $(val / exp) \% 10$.
 - Place the element in the corresponding position in radixArray based on the digit.
 - Increment the count for that digit in the counts array.
 2. **Collect Elements:**
 - Initialize a position index pos to 0.
 - For each digit (0-9):
 - For each element stored in radixArray for the current digit:

- Copy the element back to the input array starting from the pos index.
 - Increment the pos index.
 - Reset the count for the current digit to 0 in the counts array.
3. **Update Exponent:**
- Multiply exp by 10 to move to the next significant digit position (units to tens, tens to hundreds, etc.).
5. **Print the Sorted Array:**
- After all digits have been processed, the input array will be sorted.
 - Print the sorted array.

The Code

The Code

```
public class Main {
    public static void main(String[] args) {
        int[] myArray = { 170, 45, 75, 90, 802, 24, 2, 66 };

        // Step 1: Print the original array

        // Step 2: Initialize radixArray and counts
        int[][] radixArray = new int[10][myArray.length];
        int[] counts = new int[10];

        // Step 3: Find the maximum value in the array
        int maxVal = findMax(myArray);

        // Step 4: Initialize exponent to 1 (units place)
        int exp = 1;

        // Step 5: Perform sorting by each digit position
        while (maxVal / exp > 0) {
            // Step 5.1: Distribute elements into radixArray based on current digit
            for (int val : myArray) {
                int radixIndex = (val / exp) % 10;
                radixArray[radixIndex][counts[radixIndex]] = val;
                counts[radixIndex]++;
            }
        }
    }
}
```

```

// Step 5.2: Collect elements back into myArray in sorted order
int pos = 0;
for (int i = 0; i < 10; i++) {
    for (int j = 0; j < counts[i]; j++) {
        myArray[pos] = radixArray[i][j];
        pos++;
    }
    counts[i] = 0; // Reset counts for the next digit
}

// Step 5.3: Move to the next significant digit
exp *= 10;
}

// Step 6: Print the sorted array

// Method to find the maximum value in the array
}

```