TD01 Corrected - Information Retrieval and Algorithms

Third Year ISIL - University of M'Sila

Exercise 01 – Information Retrieval (Correction)

1. Information Retrieval (IR) is mainly concerned with: **Correct answer: (b)**
2. Which type of data is mainly handled by Information Retrieval systems? **Correct answer: (c)**
3. Which of the following is an example of an IR system? **Correct answer: (c)**
4. In Information Retrieval, a document can be: **Correct answer: (c)**
5. What is a query in IR? **Correct answer: (b)**
6. Relevance in Information Retrieval is: **Correct answer: (c)**
7. What is the main purpose of indexing? **Correct answer: (c)**
8. Ranking in IR systems is important because: **Correct answer: (b)**
9. Information Retrieval differs from Database systems because IR: **Correct answer: (c)**
10. The main goal of an IR system is to: **Correct answer: (c)**
11. Which component is responsible for ordering results? **Correct answer: (b)**
12. Information need refers to: **Correct answer: (b)**

Exercise 02 – Data Structures & Algorithms (Solutions)

02. Array Frequency Problem

Given array:

$$A = \{15, 25, 36, 1, 1, 25, 63, 8, 8, 8, 1, 1, 1, 8, 9, 8, 96, 8\}$$

1. Value of m

The value of m is the maximum frequency of an element. which is $m = 6$

2. What does the code do?

The code computes the maximum frequency of any element in the array.

3. Complexity

$$O(n^2)$$

4. Better Algorithm

Using sorting first:

$$O(n \log n)$$

—

03. Extract Common Values in Two Sorted Lists

Algorithm (Linear Time)

```
void extractCommon(int A[], int n, int B[], int m) {
    int i = 0, j = 0;

    while (i < n && j < m) {
        if (A[i] == B[j]) {
            printf("%d ", A[i]);
            i++;
            j++;
        }
        else if (A[i] < B[j]) {
            i++;
        }
        else {
            j++;
        }
    }
}
```

Complexity: $O(n + m)$

04. Move Zeros to End While Maintaining Order

```
void moveZeros(int arr[], int n) {
    int count = 0;

    for (int i = 0; i < n; i++) {
        if (arr[i] != 0) {
            arr[count++] = arr[i];
        }
    }

    while (count < n) {
        arr[count++] = 0;
    }
}
```

Complexity: $O(n)$

—

05. Merge Two Sorted Arrays In Place

Assume array A has enough space to hold elements of B.

```
void merge(int A[], int n, int B[], int m) {
    int i = n - 1;
    int j = m - 1;
    int k = n + m - 1;

    while (j >= 0) {
        if (i >= 0 && A[i] > B[j]) {
            A[k--] = A[i--];
        } else {
            A[k--] = B[j--];
        }
    }
}
```

Complexity: $o(n + m)$

06. Display Distinct Values in Sorted Array

```
void displayDistinct(int arr[], int n) {
    if (n == 0) return;

    printf("%d ", arr[0]);

    for (int i = 1; i < n; i++) {
        if (arr[i] != arr[i - 1]) {
            printf("%d ", arr[i]);
        }
    }
}
```

Complexity: $O(n)$

07. Find All Duplicates in Array (O(n), No Extra Space)

Constraint: $1 \leq nums[i] \leq n$

```
void findDuplicates(int nums[], int n) {
    for (int i = 0; i < n; i++) {
        int index = abs(nums[i]) - 1;

        if (nums[index] < 0) {
            printf("%d ", abs(nums[i]));
        } else {
            nums[index] = -nums[index];
        }
    }
}
```

Idea:
Use the array indices as markers by changing signs.
Complexity:

$$O(n)$$

Space Complexity:

$$O(1)$$

—