USEFUL ALGORITHMS

Using these algorithms, or a combination of them will help you during problem-solving activities, when answering past papers, and when creating your solution for the IA.

The examples demonstrate one way to carry out each task, however, in programming, there is always multiple ways to solve a problem. Give a dozen programmers a task to solve and many will arrive at the solution using a different method.

Note: This document is a work in progress and will be updated/edited as the year progresses.
All code sample can be downloaded from GitHub.

More study materials can be found at www.ibcompsci.net

All code sample can be downloaded from GitHub.

Table of Contents

Input and output content of an array

Populate an array with random numbers

The following method populates an array with random numbers.

```
import java.util.Random;                                    // Import the random number library

…
…

public void populateArray(int[] array) {
        Random random = new Random();                       // Create a new random number object
        for (int i = 0; i < array.length; i++) {            // loop through each element of the array
                array[i] = random.nextInt(100);             // and place a random number in each element.
                System.out.println(myArray[i]);             // Output each element
        }
}
```

Populate an array with user entered numbers

The following method populates an array with number entered by the user.

```
import java.util.Scanner;                                   // import the Scanner library

…
…

public void populateArray(int[] array) {

        Scanner in = new Scanner(System.in);                // create a new Scanner object

        for (int i = 0; i < array.length; i++) {            // loop through each element of the array
                System.out.print("Enter a number: ");       // prompt the user to enter a number
                array[i] = in.nextInt();                    // assign each user input to each element
        }

        // Output the contents of the array
         for (int i = 0; i < myArray.length; i++) {
                System.out.println(myArray[i]);
        }
}
```

Output the contents of an array

The following method outputs the content of an array using a loop.

```
public void outputArray(int[] array) {                  // method which accepts an array
        for (int i = 0; i < array.length; i++) {        // loop through each element of the array
                System.out.println(array[i]);           // output each element of the array
        }
}
```

You can also check if the array is empty first.

```
public void outputArray(int[] array) {
        for (int i = 0; i < myArray.length; i++) {
                if (myArray.length == 0) {              // Check if array is empty
                        System.out.println("myArray is empty.");
                } else { // check if array is empty before outputting
                        System.out.println(myArray[i]);
                }
        }
}
```

Finding items in an array

Find the smallest number in an array

The following method outputs the lowest number in an array.

```
public void getMinNumber(int[] array) {         // method that accepts an array
        int minNum = array[0];                  // create variable to hold lowest number
        for (int i = 0; i < array.length; i++) {        // loop through each element of the array
                if (array[i] < minNum) {        // current element lower than current lowest number?
                        minNum = array[i];      // if so, overwrite contents minNum
                }
        }
        System.out.println("The lowest number is " + minNum);
}
```

Find the maximum number in an array

The following method outputs the highest number in an array.

```
public void getMaxNumber(int[] array) {          // method that accepts an array
        int maxNum = 0;                          // create variable to hold highest number
        for (int i = 0; i < array.length; i++) { // loop through each element of the array
                if (array[i] > maxNum) {         // current element higher than current highest number?
                        maxNum = array[i];       // if so, overwrite contents maxNum
                }
        }
        System.out.println("The highest number is " + maxNum);          // output highest number
}
```

Find the average number in an array

The following method outputs the average number in an array.

```
public void getAverageNumber(int[] array) {              // method that accepts an array
        double averageNum = 0;
        for (int i = 0; i < array.length; i++) {         // loop through each element of the array
                averageNum = averageNum + array[i];      // add up contents of the array
        }
        averageNum = averageNum/array.length;                          // calculate the average
        System.out.println("The average number is " + averageNum);     // output average
}
```

Find a specific item in an array

The following method counts how many times a number occurs in an array.

```
public void numberCount(int[] array, int number) {       // method that accepts an array and one integer
        int count = 0;                                   // to also keep track of occurrences
        for (int i = 0; i < array.length; i++) {         // loop through each element of the array
                if (array[i] == number) {                // does current element match number?
                        count++;
                }
        }
        System.out.print("There is " + count + " occurrences of the number " + number + " in the
array");
}
```

The following method counts how many times a name occurs in an array.

Note the **.equals** instead of **==**, which is used for numerical data.

```
public void nameCount(String[] array, String name) {      // method that accepts an array and one String
        int count = 0;                                    // to also keep track of occurrences
        for (int i = 0; i < array.length; i++) {          // loop through each element of the array
                if (array[i].equals(name)) {              // does current element match name?
                        count++;
                }
        }
        System.out.print("There is " + count + " occurrence(s) of the name " + name + " in the array");
}
```

Sorting items in an array

Search YouTube for visual representations of each of these sorting algorithms. The Bubble sort algorithm often comes up in the exam.

Bubble sort

The following demonstrates the Bubble sort algorithm.

```
public void bubbleSort(int array[]) {                     // method that accepts an array

        int n = array.length;                             // assign length of array to an integer variable

        for (int i = 0; i < n-1; i++) {
                for (int j = 0; j < n-i-1; j++) {
                        if (array[j] > array[j+1]) {
                                int temp = array[j];              // assign element zero to temp variable
                                array[j] = array[j+1];            // assign element one to element zero
                                array[j+1] = temp;                // assign temp variable to element one
                        }
                }
        }
}
```