

Edith Cowan University  
CSP2348  
Data Structures  
Assignment 2

Martin Ponce  
Student 10371381

Tutor: Jitian Xiao

April 25, 2015

Contents

1 Introduction 3

2 Arrays 4

2.1 Sorting . . . . . 4

2.1.1 Insertion sort algorithm . . . . . 4

2.1.2 Insertion sort Java method . . . . . 5

2.1.3 Insertion sort analysis . . . . . 5

References 6

# 1 Introduction

This report examines array, singly-linked-list and binary tree data structures and algorithms to interact with them. Implementation of these data structures and algorithms in Java will be outlined in the report, along with analysis of algorithms used.

The array data structure will be demonstrated through the implementation of a simple lotto game. The lotto game allows up to 1000 players, with each player picking six unique integers, between 1 and 45. Each player and their respective lotto tickets are represented inside a two-dimensional array, while the winning numbers are represented inside a one-dimensional array. Insertion sort is used to sort player tickets and winning numbers arrays, while binary search and an adaptation of an array merge algorithm is used to determine if a player's ticket contains the winning numbers.

## 2 Arrays

The array data structure is demonstrated through the implementation of a simple lotto game. The lotto game allows up to 1000 players, with each player picking six unique integers, between 1 and 45, which makes up their lotto ticket. Each player and their respective lotto tickets are represented inside a two-dimensional array, while the winning numbers are represented inside a one-dimensional array.

The following classes have been created to represent the lotto game:

- **Main**: The executable class, contains `main()` method
- **PlayerTickets**: Generates a two-dimensional array which represents each player, and their picks for the lotto ticket
- **WinningNumbers**: Generates a one-dimensional array which represents the winning numbers for the lotto game
- **WinningPlayers**: Contains logic to determine who the winning players are
- **Sorter**: Helper class to sort **PlayerTickets** and **WinningNumbers** arrays
- **Randomizer**: Helper class to generate random numbers for each player pick and winning number

### 2.1 Sorting

In order to use more efficient search algorithms such as binary search, the arrays must be sorted first. The **Sorter** class contains `insertionSort()` which is called to sort elements in an array in descending order using insertion sort algorithm. Implementation is shown in Java code 2.1.

#### 2.1.1 Insertion sort algorithm

To sort `a[left...right]` into ascending order:

1. For  $r = \text{left} + 1, \dots, \text{right}$ , repeat:
  - 1.1. Let  $\text{val} = a[r]$
  - 1.2. Insert  $\text{val}$  into its correct sorted position in `a[left...r]`
2. Terminate

(Watt & Brown, 2001, p. 1)

### 2.1.2 Insertion sort Java method

```
1 private static final int[] insertionSort(int[] array) {
2
3     // 1.0 For r = left+1, ..., right, repeat:
4     for(int i = 1; i < array.length; i++) {
5
6         // 1.1 Let val = a[r] (replaced with key)
7         int key = array[i];
8
9         // 1.2 Insert val into a[left...r] in such a way that
10        // the subarray is sorted
11
12        // j tracks the sorted invariant
13        int j = i - 1;
14
15        // while j >= 0 and array[j] > key,
16        while(j >= 0 && array[j] > key) {
17
18            // set array[j + 1] to array[j]
19            array[j + 1] = array[j];
20            // decrement j
21            j--;
22        }
23
24        // after inner loop, set array[j + 1] to key
25        array[j + 1] = key;
26    }
27
28    // 2.0 Terminate
29    return array;
30 }
```

Java code 2.1: Insertion sort method

### 2.1.3 Insertion sort analysis

## References

Watt, D. A., & Brown, D. F. (2001). *Java Collections - An Introduction to Abstract Data Types, Data Structures and Algorithms* (1st ed.). New York: John Wiley & Sons.