Sohan Chatterjee

Lab 11 Complexity Analysis

SSW 315

14 November 2022

Complexity Analysis #1 - Date Helper

Method	Worst Case	Storage Complexity	Justification for non-constant complexities
getMonth(int month)	none	O(1)	
getMonth(String month)	n=”september”	O(n)	The method splits the string into characters and compares each character to the characters of a default string through a for-loop.
getDay(int day)	none	O(1)	
getDay(String day)	n=”wednesday”	O(n)	The method splits the string into characters and compares each character to the characters of a default string through a for-loop.
getDate(int year, int month, int day)	none	O(1)	
getDate(String date)	n=10	$O(n^2)$	The String substring method has a linear complexity as does the Integer parseInt method which makes the overall complexity n^2.
getIntegerValue(String input)	n.length()	O(n)	Integer’s parseInt function has a complexity of n because it iterates through a given string.
isLeapYear(int year)	none	O(1)	
isValidDate(int year, int month, int day)	none	O(1)	
dayOfYear(int year, int month, int day)	n=month	O(n)	The loop within the method runs as many times as there are months and calls a constant function within the

			class, giving it a linear complexity.
daysOfMonth(int year, int month)	none	O(1)	
dayOfWeek(int year, int month, int day)	n=year OR n=month	O(n)	This method calls the daysBetween method which has a complexity of O(n).
daysBetween(int yearFirst, int monthFirst, int dayFirst, int yearSecond, int monthSecond, int daySecond)	n=yearFirst OR n=monthFirst	O(n)	There are two while loops which contribute to this complexity, both being linear as the loops run until both of the values within each condition are equal. This gives an overall complexity of O(n + n), which can be simplified to O(n).
randomYear(), randomYear(int end), randomYear(int start, int end)	none	O(1)	
randomMonth()	none	O(1)	
randomDay(), randomDay(int month), randomDay(int year, int month)	none	O(1)	
randomDate()	none	O(1)	

```
//Sohan Chatterjee
//SSW 315 Date Helper
//September 21, 2022
import java.util.Random;

import javax.lang.model.util.ElementScanner14;

import java.util.Arrays;

public class DateHelper {
    public static String getMonth(int month) {
        switch (month) {
```

```
            case 1:
                return "January";
            case 2:
                return "February";
            case 3:
                return "March";
            case 4:
                return "April";
            case 5:
                return "May";
            case 6:
                return "June";
            case 7:
                return "July";
            case 8:
                return "August";
            case 9:
                return "September";
            case 10:
                return "October";
            case 11:
                return "November";
            case 12:
                return "December";
        }
        return "";
    }

    public static int getMonth(String month) {
        for (int i = 2; i < month.length(); i++) {
            if (month.substring(0,
i).equalsIgnoreCase("january".substring(0, i)))
                return 1;
            else if (month.substring(0,
i).equalsIgnoreCase("february".substring(0, i)))
                return 2;
            else if (month.substring(0,
i).equalsIgnoreCase("march".substring(0, i)))
                return 3;
```

```
            else if (month.substring(0,
i).equalsIgnoreCase("april".substring(0, i)))
                return 4;
            else if (month.substring(0,
i).equalsIgnoreCase("may".substring(0, i)))
                return 5;
            else if (month.substring(0,
i).equalsIgnoreCase("june".substring(0, i)))
                return 6;
            else if (month.substring(0,
i).equalsIgnoreCase("july".substring(0, i)))
                return 7;
            else if (month.substring(0,
i).equalsIgnoreCase("august".substring(0, i)))
                return 8;
            else if (month.substring(0,
i).equalsIgnoreCase("september".substring(0, i)))
                return 9;
            else if (month.substring(0,
i).equalsIgnoreCase("october".substring(0, i)))
                return 10;
            else if (month.substring(0,
i).equalsIgnoreCase("november".substring(0, i)))
                return 11;
            else if (month.substring(0,
i).equalsIgnoreCase("december".substring(0, i)))
                return 12;
            else
                return 0;
        }
        return 0;
    }

    public static String getDay(int day) {
        switch (day) {
            case 0:
                return "Sunday";
            case 1:
                return "Monday";
            case 2:
```

```
                return "Tuesday";
            case 3:
                return "Wednesday";
            case 4:
                return "Thursday";
            case 5:
                return "Friday";
            case 6:
                return "Saturday";
            case 7:
                return "Sunday";
        }
        return "";
    }

    public static int getDay(String day) {
        for (int i = 2; i < day.length(); i++) {
            if (day.substring(0, i).equalsIgnoreCase("monday".substring(0,
i)))
                return 1;
            else if (day.substring(0,
i).equalsIgnoreCase("tuesday".substring(0, i)))
                return 2;
            else if (day.substring(0,
i).equalsIgnoreCase("wednesday".substring(0, i)))
                return 3;
            else if (day.substring(0,
i).equalsIgnoreCase("thursday".substring(0, i)))
                return 4;
            else if (day.substring(0,
i).equalsIgnoreCase("friday".substring(0, i)))
                return 5;
            else if (day.substring(0,
i).equalsIgnoreCase("saturday".substring(0, i)))
                return 6;
            else if (day.substring(0,
i).equalsIgnoreCase("sunday".substring(0, i)))
                return 7;
        }
        return 0;
```

```
    }

    public static String getDate(int year, int month, int day) {
        if (year < 0 || month < 0 || month > 12 || day < 0 || day > 31)
            return "";
        String yearString = "" + year;
        String monthString = "" + month;
        String dayString = "" + day;
        if (month < 10)
            monthString = "0" + month;
        if (day < 10)
            dayString = "0" + day;
        return yearString + "-" + monthString + "-" + dayString;
    }

    public static int[] getDate(String date) {
        int[] dateArray = new int[3];
        if (date.length() == 10) {
            dateArray[0] = getIntegerValue(date.substring(0, 4));
            dateArray[1] = getIntegerValue(date.substring(5, 7));
            dateArray[2] = getIntegerValue(date.substring(8, 10));
        } else {
            dateArray[0] = 0;
            dateArray[1] = 0;
            dateArray[2] = 0;
        }
        if (dateArray[0] == -1 || dateArray[1] == -1 || dateArray[2] ==
-1) {
            dateArray[0] = 0;
            dateArray[1] = 0;
            dateArray[2] = 0;
        }
        System.out.println(Arrays.toString(dateArray));
        return dateArray;
    }

    public static int getIntegerValue(String input) {
        int value;
        try {
            value = Integer.parseInt(input);
```

```
        } catch (NumberFormatException e) {
            value = -1;
        }
        return value;
    }

    public static boolean isLeapYear(int year) {
        if ((year % 400) == 0)
            return true;
        else if (((year % 4) == 0) && ((year % 100) != 0))
            return true;
        else
            return false;
    }

    public static boolean isValidDate(int year, int month, int day) {
        if (year < 0 || month < 0 || month > 12 || day < 0 || day > 31)
            return false;
        return true;
    }

    public static int dayOfYear(int year, int month, int day) {
        int totalDays = 0;
        for (int i = 1; i < month; i++) {
            totalDays += daysOfMonth(year, i);
        }
        totalDays += day;
        return totalDays;
    }

    public static int daysOfMonth(int year, int month) {
        switch (month) {
            case 1:
                return 31;
            case 2:
                if (isLeapYear(year))
                    return 29;
                return 28;
            case 3:
                return 31;
```

```
            case 4:
                return 30;
            case 5:
                return 31;
            case 6:
                return 30;
            case 7:
                return 31;
            case 8:
                return 31;
            case 9:
                return 30;
            case 10:
                return 31;
            case 11:
                return 30;
            case 12:
                return 31;
        }
        return 0;
    }

    public static String dayOfWeek(int year, int month, int day) {
        final int YEAR = 1;
        final int MONTH = 1;
        final int DAY = 1;
        return getDay((daysBetween(YEAR, MONTH, DAY, year, month, day) %
7) + 1);
    }

    public static int daysBetween(int yearFirst, int monthFirst, int
dayFirst, int yearSecond, int monthSecond,
            int daySecond) {
        int totalDays = 0;
        while (yearFirst != yearSecond) {
            if (isLeapYear(yearFirst))
                totalDays += 366;
            totalDays += 365;
            yearFirst++;
        }
```

```
        while (monthFirst != monthSecond) {
            totalDays += daysOfMonth(yearSecond, monthFirst);
            monthFirst++;
        }
        totalDays += daySecond - dayFirst;
        return totalDays;
    }

    public static int randomYear(int start, int end) {
        Random r = new Random();
        return r.nextInt(end - start) + start;
    }

    public static int randomYear(int end) {
        return randomYear(1, end);
    }

    public static int randomYear() {
        return randomYear(1, 9999);
    }

    public static int randomMonth() {
        Random r = new Random();
        return r.nextInt(12) + 1;
    }

    public static int randomDay(int year, int month) {
        Random r = new Random();
        return r.nextInt(daysOfMonth(year, month)) + 1;
    }

    public static int randomDay(int month) {
        Random r = new Random();
        return r.nextInt(daysOfMonth(2022, month)) + 1;
    }

    public static int randomDay() {
        Random r = new Random();
        return r.nextInt(31) + 1;
    }
```

```
    public static int[] randomDate() {
        int[] randomDate = { randomYear(), randomMonth(), randomDay() };
        System.out.println(Arrays.toString(randomDate));
        return randomDate;
    }
}
```