/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Assignment 3

\* Name: Siyan Wang

\* Email: swang2@haverford.edu

\* Course: CS 206

\* Submitted: 9/24/2015

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Reflection：

This code is much more complicated than the code we wrote before. It relates to two classes and several methods. I think to write a structural tree before writing the code is very helpful. It can help me know what methods and what classes are necessary to include in my codes. At the same time, I feel it is very necessary to click “tap” for each line so that the structure will be clearer. Also, think as many situations as possible is important. At first, I input “ 19041”(with a space before 1) and I found it is not ok to get an output. Hence, I add a trim sentence to solve this problem and it is proved to work.

Console window:

weiphonedeMacBook-Air:desktop weiphone$ javac Search.java

weiphonedeMacBook-Air:desktop weiphone$ java Search

You asked me to search for zip code

The input is not valid. Please enter a new one.

Do you want to search again? Yes

You asked me to search for zip code 123

The input is not valid. Please enter a new one.

Do you want to search again? Yes

You asked me to search for zip code abc

The input is not valid. Please enter a new one.

Do you want to search again? Yes

You asked me to search for zip code 19041

The time linear search needs is 4456390

The time binary search needs is 122866

The Zip code 19041 belongs to Haverford, PA

Do you want to search again? Yes

You asked me to search for zip code 97546

The time linear search needs is 13228779

The time binary search needs is 7843

The zip code 97546 does not exist.

Do you want to search again? Yes

You asked me to search for zip code 94567

The time linear search needs is 10288561

The time binary search needs is 6806

The Zip code 94567 belongs to Pope Valley, CA

Do you want to search again? Yes

You asked me to search for zip code abcde

The input is not valid. Please enter a new one.

Do you want to search again? Yes

You asked me to search for zip code 19 041

The input is not valid. Please enter a new one.

Do you want to search again? Yes

You asked me to search for zip code 19041

The time linear search needs is 631005

The time binary search needs is 12460

The Zip code 19041 belongs to Haverford, PA

Do you want to search again? Yes

You asked me to search for zip code bifwidanxjkbv

The input is not valid. Please enter a new one.

Do you want to search again? Yes

You asked me to search for zip code 123d4

The input is not valid. Please enter a new one.

Do you want to search again? Yes

You asked me to search for zip code 199999

The input is not valid. Please enter a new one.

Do you want to search again? No

Goodbye! Thanks:)

weiphonedeMacBook-Air:desktop weiphone$ java Search

You have cancelled the search.

Goodbye! Thanks:)

Code:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Assignment 3

\* Name: Siyan Wang

\* Email: swang2@haverford.edu

\* Course: CS 206

\* Submitted: 9/29/2015

\*

\* The source code for zip code and town name match.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

import java.util.Arrays;

import javax.swing.JOptionPane;

public class Search{

//Input the data

public static Place[] inputData(File file) throws FileNotFoundException{

//Access to the file

Scanner lines= new Scanner(file);

//get the basic info of the file

String[] info= lines.nextLine().split(",");

int length = Integer.parseInt(info[0].split(" ")[1]);

//Create an array of Place objects

Place[] data= new Place[length];

int counter = 0;

//Put each place into the array

while(lines.hasNext()){

String[] line=lines.nextLine().split("\t");

String townState= line[3];

String[] townState\_split=townState.split(", ");

data[counter]=new Place(line[0], townState\_split[0], townState\_split[1]);

counter++;

}

lines.close();

return data;

}// end of inputData

//Check whether a value is a integer.

public static boolean isInteger(String value) {

try {

Integer.parseInt(value);

return true;

} catch (NumberFormatException e) {

return false;

}

}//end of isInteger

//Use linear search to get the location of a zip

public static int searchData(String input, Place[] dataPiece){

//Prepare for the search

int length= dataPiece.length;

//The input must be a 5-digit number

if(input.length() !=5)

throw new NumberFormatException();

//Check whether the input is an integer;

if (isInteger(input)== false)

throw new NumberFormatException();

//Test if the input is a US zip code(with linear search)

long startTime = System.nanoTime();

for (int i=0; i<length; i++) {

if (dataPiece[i].equals(input)){

long endTime = System.nanoTime();

long duration = (endTime - startTime);//calculate the time;

System.out.println("The time linear search needs is "+ duration);

return i;

}

}

long endTime = System.nanoTime();

long duration = (endTime - startTime);//calculate the time;

System.out.println("The time linear search needs is "+ duration);

return -1;

}//end of searchData

//use the binary search the index

public static int binarySearch(String input2, Place[] data2){

Arrays.sort(data2);

//The input must be a 5-digit number;

if(input2.length() !=5)

throw new NumberFormatException();

//Check whether the input is an integer;

if (isInteger(input2)== false)

throw new NumberFormatException();

//Use binary search to find where the zip code is;

Place dataPlace =new Place(input2, null, null);

long startTime2 = System.nanoTime();

int index=Arrays.binarySearch(data2, dataPlace);

long endTime2 = System.nanoTime();

long duration2 = (endTime2 - startTime2);//calculate the time;

System.out.println("The time binary search needs is "+ duration2);

return index;

}//end of binarySearch

public static void main(String[] args){

int response=0;

do{

try{

//import the data

Place[] data = inputData(new File("data/zips.txt"));

//Ask for input

String input = JOptionPane.showInputDialog(null, "Please enter the zip code you want to search for: ","Zip Code Search", JOptionPane.QUESTION\_MESSAGE);

if (input !=null){//and click ok certainly..

//Eliminate the spaces before and after the input first.

input=input.trim();

System.out.println("You asked me to search for zip code "+ input);

//search the location

int linearIndex=searchData(input, data);

int binaryIndex=binarySearch(input, data);

if (binaryIndex>=0){//click OK of course...a valid zip code...

Place locationInfo= data[binaryIndex];

String outputTown= locationInfo.getTown();

String outputState = locationInfo.getState();

JOptionPane.showMessageDialog(null, "The Zip code "+ input+ " belongs to "+outputTown+", "+outputState);

System.out.println("The Zip code "+ input+ " belongs to "+outputTown+", "+outputState);

//Continue or not

String[] options={"Yes", "No"};

response=JOptionPane.showOptionDialog(null, "Do you want to search again?","Zip Code Search", JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE, null, options, options[0]);

if (response ==0){//continue searching

System.out.println("Do you want to search again? Yes\n");

}else{//exit

System.out.println("Do you want to search again? No");

}

}else{//the zip code does not exist

System.out.println("The zip code "+input+" does not exist.");

//Ask if continue or not

String[] options={"Yes", "No"};

response=JOptionPane.showOptionDialog(null, "Do you want to search again?", "Zip Code Search", JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE, null, options, options[0]);

if (response==0){//continue searching

System.out.println("Do you want to search again? Yes\n");

}else{//exit

System.out.println("Do you want to search again? No");

}

}

}else{//click cancel

System.out.println("You have cancelled the search.");

response = 1;

}

}catch (FileNotFoundException e){

System.out.println("The path is not valid or cannot be found!");

System.exit(1);

}catch(NumberFormatException e){

//Give info abou the incorrect input.

JOptionPane.showMessageDialog(null, "The input is not valid. Please enter a new one.", "Input error", JOptionPane.ERROR\_MESSAGE);

System.out.println("The input is not valid. Please enter a new one.");

//Ask if continue or not

String[] options={"Yes", "No"};

response=JOptionPane.showOptionDialog(null, "Do you want to search again?", "Zip Code Search", JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE, null, options, options[0]);

if (response==0){//continue searching

System.out.println("Do you want to search again? Yes\n");

}else{//exit

System.out.println("Do you want to search again? No");

}

}

}while(response==0);

System.out.println("Goodbye! Thanks:) \n");

}//end of the main

}//end of the class

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Assignment 3

\* Name: Siyan Wang

\* Email: swang2@haverford.edu

\* Course: CS 206

\* Submitted: 9/24/2015

\*

\* The source code for Place class to searve to Search.java.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class Place implements Comparable <Place>{

//Attributes

private String zip;

private String town;

private String state;

/\*\*

\* the constructor for a place

\* @param zipcode

\* @param town

\* @param state

\*/

public Place(String zipCode, String townName, String stateName){

zip=zipCode;

town=townName;

state=stateName;

}

//ancestor method: return zip code

public String getZip(){

return zip;

}

//ancestor method: return town name

public String getTown(){

return town;

}

//ancestor method: return state name

public String getState(){

return state;

}

public String toString(){

return "The zip code "+ zip + " belongs to "+town+", "+state;

}

public Boolean equals(String seek){

return zip.equals(seek);

}

public int compareTo(Place place){

Integer placeInt=Integer.parseInt(place.getZip());

Integer existingInt=Integer.parseInt(getZip());

return existingInt.compareTo(placeInt);

}

}