2017

Electronic Telephone Directory

CSC2001F ASSIGNMENT 1

FARESA MPHEPHU ~ MPHFAR001

Electronic Telephone Directory

Searching in a telephone directory the manual way using a book is a time-consuming task compared to using the modern algorithm driven method,

Different programmers write different algorithms leading to some of those algorithms to be redundant as they can be so slow that searching for a name in a phone directory is better found manually. There are great algorithms as well such as the Binary Search algorithm which reduces entries searched by half if an item is not found

This Report will illustrate how a linear Search Performs versus a Binary Search

Application Design

The Architecture of Printit and Searchit are quite similar in the way that both initially read a datafile containing all telephone directory records, these applications make use of a class Record.java which is a data type containing `{Key: Value}` pairs of telephone directory entries

Telephone records data have been edited to contain the character ~ which acts as a separator of the data into the format: Name ~ Phone Address, this is prerequisite for creating the `{Key: Value}` parameters to create a Record datatype,

Architecture of Record.java

- String Attributes `Key` and `Value`
- Implements Comparable
- Overwrites `compareTo ()` method
- Method `toString ()` which returns the Value

When the programs start execution, a Binary Search Tree is created. Both Printit and Searchit read Telephone Directory entries from a datafile until the last line, each entry will be split by the character ~ to a temporary `String record array`, the first argument `record [0]` is the name, the second argument `record [1]` is the entire entry.

A `Binary Tree Node <Record>` which represents a telephone directory entry is created, the tree node is then inserted inside the Binary Search Tree (BST).

The differences in the two applications are commands executed after insertion into the BST is completed, Printit simply organizes the data per the name and return Record `Values` to the console.

Searchit proceeds to read queries from a `queryfile` containing only names, the names are individually passed into a method named search

Architecture of Method `search (String query, BinarySearchTree<Record> tree)`

- ¬ Returns a Record datatype
- ¬ Takes in a query and search in a Binary Search Tree
- ¬ Utilizes BinarySearchTree method `find ()` which returns a Binary Tree Node
- ¬ Creates a `notFound` Record which represents the case where `find ()` returns `null`
- ¬ Reads the data from the node (which is a Record) and returns the Value

The main (String [] args) method takes in one parameter which is the data subset and is used to vary the number of elements inside the Binary Search Tree

Printit Java has a variable `queryfile` which can be changed from queryfile to `queryfile_20` which contains only 20 elements for demonstration purposes only

How to Use / Makefile

The makefile conducts all the operations of the Electronic Telephone Directory Tasks from

Commands and description

`make`	Compiles classes to bin/
`make Printit`	**Run the Printit Application**
`make Searchit`	**Run the Searchit with dataSubset=10000**
`make SearchitLinear`	**Run SearchitLinear with dataSubset=10000**
`make experiment`	**Conducts Linear vs Binary Search experiment**
`make clean`	**Remove all class files from bin**

The first 20 lines of the output from Printit.

Abbott Alec ~ 489-848-7299 03707 Botsford Fork, Lima
Abbott Alexandria ~ 318.679.5603 x712 44812 Wilderman Mountain, Vallejo
Abbott Alia ~ 507.340.1186 76400 Barton Fields #044, Cerritos
Abbott Brando ~ 602.992.4016 02519 Zackery Village, San Mateo
Abbott Elwyn ~ 788.603.8604 88126 Bruen Common, Beverly Hills
Abbott Hosea ~ 1-035-079-0176 x61480 51832 Bayer Pass, Simi Valley
Abbott Ima ~ 823.283.2198 x7192 87191 Suite Z, Selma
Abbott Josh ~ 822.752.1004 27010 Sanford Center, Stanton
Abbott Leann ~ 516-835-0116 17296 Elta Crossroad #362, Newport Beach
Abbott Meda ~ 1-117-789-3061 18565 Suite B, Fountain Valley
Abbott Murray ~ 1-654-279-2374 22345 Runte Garden, Steubenville
Abbott Novella ~ 297-763-2822 32763 Langosh Route, San Diego
Abbott Rahsaan ~ (681)856-6604 x642 90282 Haag Keys, Garden Grove
Abbott Sadye ~ (961)238-9093 52000 Marques Loaf #288, Placentia
Abbott Santina ~ 1-515-459-1556 78469 Renner Mill, Agoura Hills
Abernathy Amparo ~ 1-052-394-1236 x29668 96179 Feil Tunnel #352, Canton
Abernathy Austyn ~ 1-486-893-0367 98827 Gerlach Pike Apt. 743, Apple Valley
Abernathy Catalina ~ 1-331-934-0147 14576 Harber Knolls, Riverside
Abernathy Chadd ~ (552)753-8320 x85031 23694 Pier F, Tempe
Abernathy Cicero ~ (637)882-6835 x72457 36296 Batz Walk, San Francisco

20 entry query file used for SearchIt and SearchItLinear,

Ruecker Bailey
Wolff Selina
Nienow Irwin
Stehr Cornell
Dibbert Eulalia
Muller Noel
Ruecker Kavon
Gottlieb Mona
Wilkinson Montana
Fahey Lane
Little Noelia
Luettgen Nathan
Barrows Jordan
Kreiger Tad
Kuhlman Alia
Green Foster
Gerhold Cristobal
Cole Tierra
Paucek Mose
Lehner Alberta

Output from Searchit.java

Ruecker Bailey ~ 1-236-279-3047 91390 Justine Camp Suite 882, Bloomington
Wolff Selina ~ 299.719.6886 x69076 82002 Blanda Forks Apt. 849, Hawthorne
Nienow Irwin ~ 641-746-9142 42061 Cordia Island Apt. 272, Newport Beach
Stehr Cornell ~ 1-819-190-5563 88939 Jamison Greens #359, West Hollywood
Dibbert Eulalia ~ (428)697-7350 x55199 13364 Floor 19, Cypress
Muller Noel ~ 1-733-623-5956 x0350 26057 Eichmann Rest, Orange
Ruecker Kavon ~ 896.609.7469 27409 Candida Estates, Tuscaloosa
Gottlieb Mona ~ 1-511-326-4258 19842 Fae Shoal Apt. 656, Youngstown
Wilkinson Montana ~ 1-331-632-4585 x07248 41777 Sonia Points, Cudahy
Fahey Lane ~ 1-220-139-2838 x649 04923 Flatley Island Suite 476, San Clemente
Little Noelia ~ 1-107-781-4467 x140 97918 Ward Fords Suite 533, San Clemente
Luettgen Nathan ~ (779)727-7731 41135 O'Hara Fort #292, Murrieta
Barrows Jordan ~ 419.600.8482 40605 Jose Divide #889, Barrow
Kreiger Tad ~ 322-975-3378 x706 56941 Haylie Knolls, Sahuarita
Kuhlman Alia ~ 063.398.9808 x87859 16990 Schoen Extension, Carson
Green Foster ~ 1-265-752-9982 x87443 81530 Lazaro Pike, Lafayette
Gerhold Cristobal ~ 1-334-679-6851 x50935 56912 Ledner Plains, Thousand Oaks
Cole Tierra ~ 813-508-5298 x0174 64136 Gerhold Drive, Sierra Vista
Paucek Mose ~ 327.239.4036 x03681 36902 Fay Harbor, San Juan Capistrano
Lehner Alberta ~ (460)301-1274 x351 67100 Schumm Pines, Barrow

Output from SearchitLinear.java

Ruecker Bailey ~ 1-236-279-3047 91390 Justine Camp Suite 882, Bloomington
Wolff Selina ~ 299.719.6886 x69076 82002 Blanda Forks Apt. 849, Hawthorne
Nienow Irwin ~ 641-746-9142 42061 Cordia Island Apt. 272, Newport Beach
Stehr Cornell ~ 1-819-190-5563 88939 Jamison Greens #359, West Hollywood
Dibbert Eulalia ~ (428)697-7350 x55199 13364 Floor 19, Cypress
Muller Noel ~ 1-733-623-5956 x0350 26057 Eichmann Rest, Orange
Ruecker Kavon ~ 896.609.7469 27409 Candida Estates, Tuscaloosa
Gottlieb Mona ~ 1-511-326-4258 19842 Fae Shoal Apt. 656, Youngstown
Wilkinson Montana ~ 1-331-632-4585 x07248 41777 Sonia Points, Cudahy
Fahey Lane ~ 1-220-139-2838 x649 04923 Flatley Island Suite 476, San Clemente
Little Noelia ~ 1-107-781-4467 x140 97918 Ward Fords Suite 533, San Clemente
Luettgen Nathan ~ (779)727-7731 41135 O'Hara Fort #292, Murrieta
Barrows Jordan ~ 419.600.8482 40605 Jose Divide #889, Barrow
Kreiger Tad ~ 322-975-3378 x706 56941 Haylie Knolls, Sahuarita
Kuhlman Alia ~ 063.398.9808 x87859 16990 Schoen Extension, Carson
Green Foster ~ 1-265-752-9982 x87443 81530 Lazaro Pike, Lafayette
Gerhold Cristobal ~ 1-334-679-6851 x50935 56912 Ledner Plains, Thousand Oaks
Cole Tierra ~ 813-508-5298 x0174 64136 Gerhold Drive, Sierra Vista
Paucek Mose ~ 327.239.4036 x03681 36902 Fay Harbor, San Juan Capistrano
Lehner Alberta ~ (460)301-1274 x351 67100 Schumm Pines, Barrow

Experiment Design

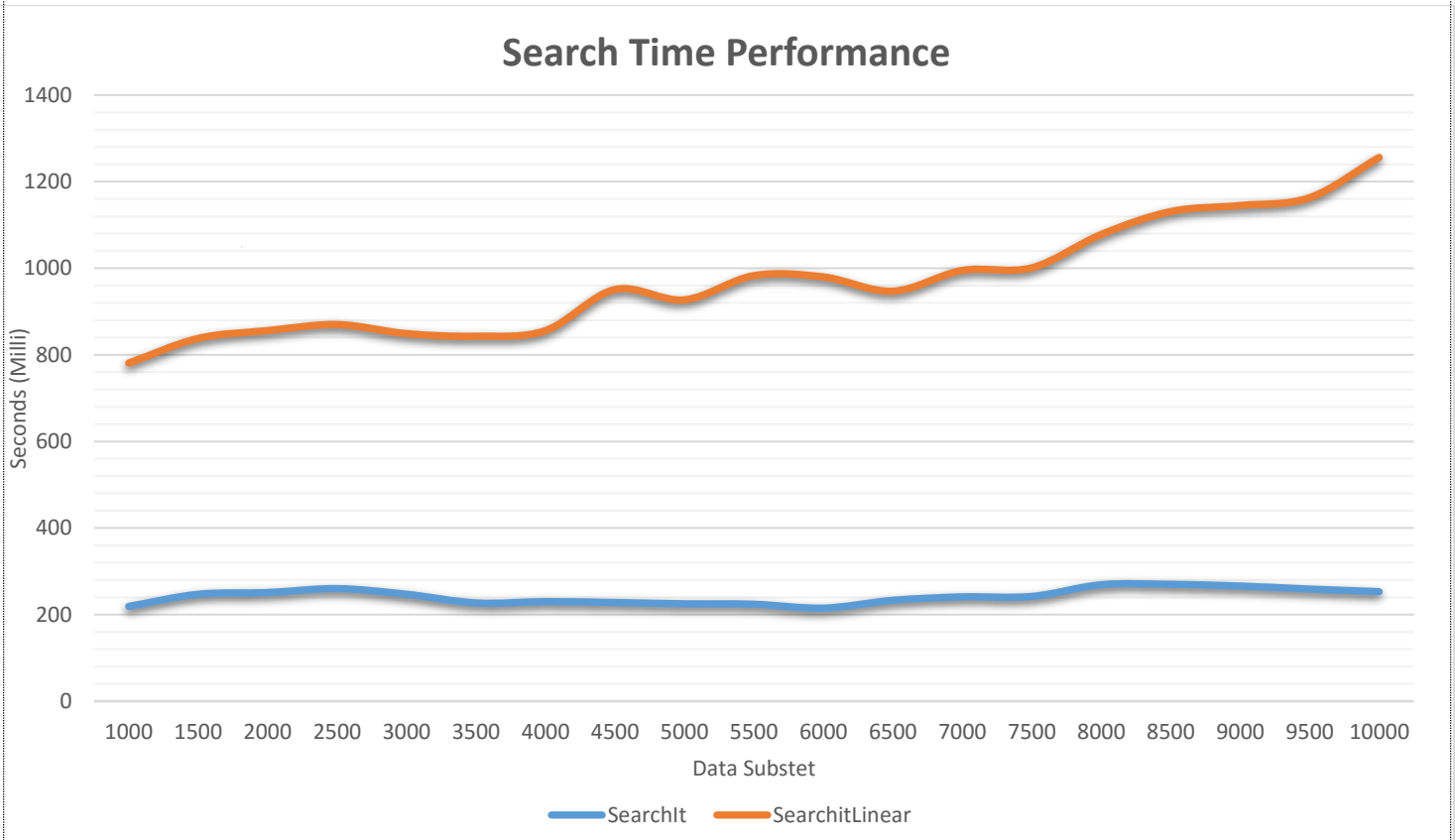
A bash script called “runeExperiment.sh” runs the experiment automatically, the bash script can be called by the command make experiment

runExperiment.sh conducts the following tasks:

1. Runs a loop from 0 to 10001 with a step of 500 using variable dataSubset
2. The data subset is passed inside the main method of both Searchit and SearchitLinear
3. .Displays the current `dataSubset` being tested
4. runs `time java Searchit $dataSubset` then sleeps for 5 seconds to give me time to record the data into a graph
5. runs `time java SearchitLinear $dataSubset` then sleeps
6. repeats process 3 to 5 until the dataSubset is 10000

for more accurate reading, `runExperiment. sh` was executed multiple times and the average three times were recorded manually, the time used to measure the execution rum time was Real time s it best simulates how long it would take an algorithm to find records in the real word scenario

the file experiment_queryfile contains the first 1000 elements to be searched when conducting the experiment meaning that the base case for the dataSubset = 1000 and then research with a step of 500



X-Values	SearchIt	SearchitLinear
1000	219	562
1500	247	591
2000	251	605
2500	260	610
3000	247	602
3500	227	616
4000	230	626
4500	228	723
5000	225	702
5500	224	759
6000	215	765
6500	233	714
7000	241	754
7500	242	759
8000	269	809
8500	270	861
9000	266	879
9500	259	904
10000	253	1003

Discussion

The initial queryfile contains 500 entries, the base case will be dataSubset = 500 and this represents a case where all the names have been found in a database that has only those names to search through

One major characteristic of Binary Search vs Linear Search is

Searchit performed significantly better than SearchitLinear throughout the experiment, as the data subsets increased, Binary search was increasing at a rate of O(logN) due to the fact that the elements to search through get haled every time an element is not found, whereas for SearchitLinear , if an item is not found , it still has n number of entries to search through so the time complexity of SearchitLinear of order O(N)

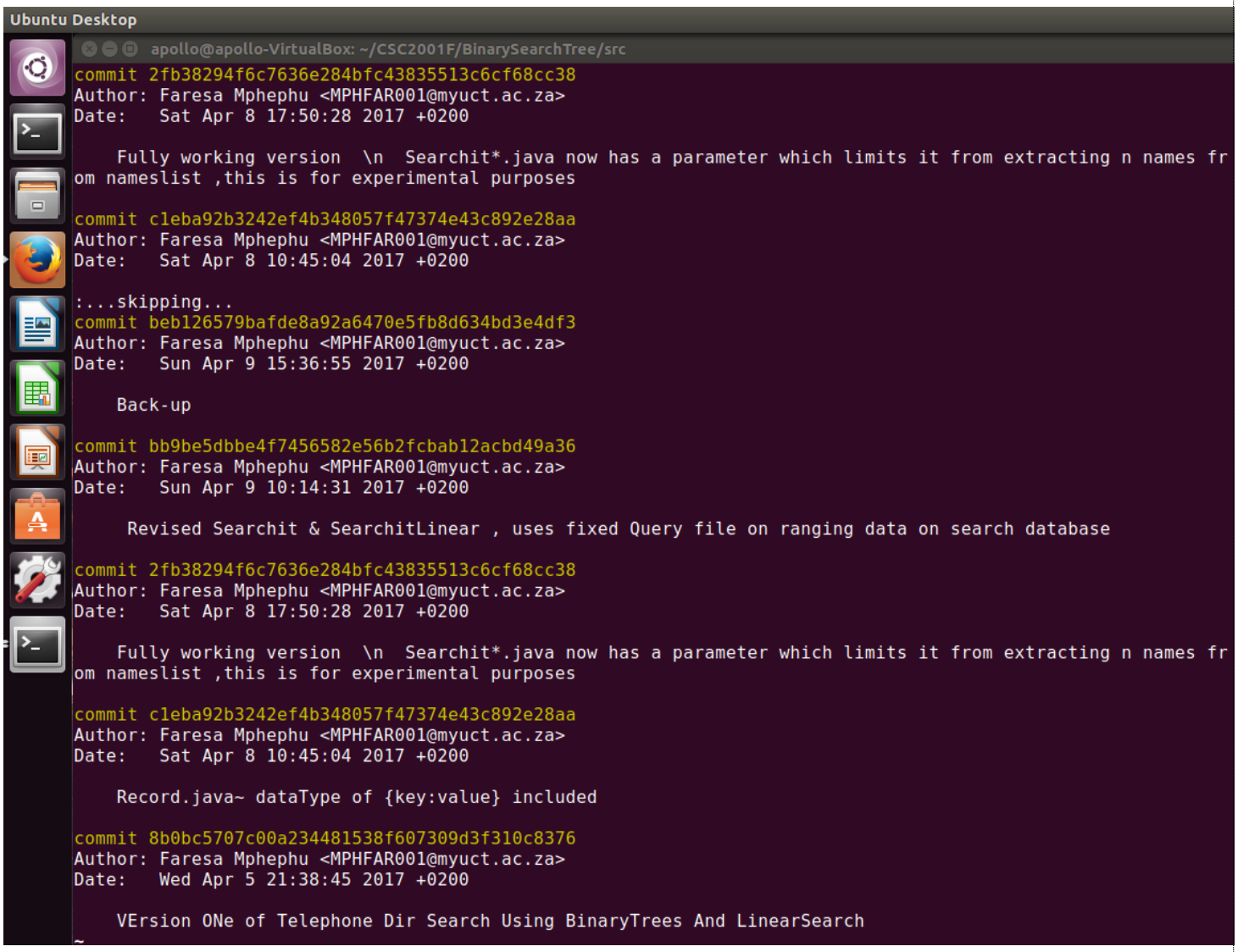
SearchitLinear recorded higher time differences from the previous dataSubset = n , whereas Binary Search increases time but not enough to notice a significant time difference between testing for dataSubset =n and previous dataSubset , Binary Search Almost looks like its constant , but repeated tests and an increment in the number of queries searched for from 1000 to 5000 or 10000 show that the binary Search increases complexity as data subsets increase , noticeable changes in time difference for binary search occur when the data being searched for is doubled .

Conclusion

Binary search tree out performs SearchitLinear in finding elements, however, Searchit has is downside that insertion and deletion can be complex since the data structure needs to remain ordered after insertion or deletion, SearchitLinear is an inefficient algorithm for searching tasks

Git

Git was useful from the onset of the project thought-out , initially the algorithms of the code were partially correct due to the fact that I thought the experiment was based on searching the Binary Search tree with ranging data subsets of queryfile (name list only) , when I realised that this cannot be the right method of determining algorithm speed it was easy to trace back my code and check out the branch which contained the algorithms from the point where I lost tract of the Assignment scope


Ubuntu Desktop
apollo@apollo-VirtualBox: ~/CSC2001F/BinarySearchTree/src
commit 2fb38294f6c7636e284bfc43835513c6cf68cc38
Author: Faresa Mphephu <MPHFAR001@myuct.ac.za>
Date: Sat Apr 8 17:50:28 2017 +0200
Fully working version \n Searchit*.java now has a parameter which limits it from extracting n names fr
om nameslist ,this is for experimental purposes
commit c1eba92b3242ef4b348057f47374e43c892e28aa
Author: Faresa Mphephu <MPHFAR001@myuct.ac.za>
Date: Sat Apr 8 10:45:04 2017 +0200
:...skipping...
commit beb126579bafde8a92a6470e5fb8d634bd3e4df3
Author: Faresa Mphephu <MPHFAR001@myuct.ac.za>
Date: Sun Apr 9 15:36:55 2017 +0200
Back-up
commit bb9be5dbbe4f7456582e56b2fcbab12acbd49a36
Author: Faresa Mphephu <MPHFAR001@myuct.ac.za>
Date: Sun Apr 9 10:14:31 2017 +0200
Revised Searchit & SearchitLinear , uses fixed Query file on ranging data on search database
commit 2fb38294f6c7636e284bfc43835513c6cf68cc38
Author: Faresa Mphephu <MPHFAR001@myuct.ac.za>
Date: Sat Apr 8 17:50:28 2017 +0200
Fully working version \n Searchit*.java now has a parameter which limits it from extracting n names fr
om nameslist ,this is for experimental purposes
commit c1eba92b3242ef4b348057f47374e43c892e28aa
Author: Faresa Mphephu <MPHFAR001@myuct.ac.za>
Date: Sat Apr 8 10:45:04 2017 +0200
Record.java~ dataType of {key:value} included
commit 8b0bc5707c00a234481538f607309d3f310c8376
Author: Faresa Mphephu <MPHFAR001@myuct.ac.za>
Date: Wed Apr 5 21:38:45 2017 +0200
VErsion ONe of Telephone Dir Search Using BinaryTrees And LinearSearch
~

END