

A performance comparison of search trees

Business Analysis

Przemysław Rosiński
Piotr Janaszek

Faculty of Mathematics and Information Science
Warsaw University of Technology

20 October 2011

DOCUMENT METRIC

Project:	A performance comparison of search trees	Company:	WUT
Name:	Business Analysis		
Topics:	Requirements, user paths, sources of data		
Author:	Przemysław Rosiński, Piotr Janaszek		
File:	SearchTreesPerformance_BA.pdf		
Version no:	1.1	Status:	Working
		Opening date:	10.10.2011
Summary:	Client requirements for the application		
Authorized by:		Last modification date:	20.10.2011

HISTORY OF CHANGES

Version	Date	Author	Description
0.0.1	10.10.2011	Przemysław Rosiński	Created document
0.2	10.10.2011	Przemysław Rosiński, Piotr Janaszek	Created Preface For Management
0.4	12.10.2011	Przemysław Rosiński, Piotr Janaszek	Created Requirements
0.4.5	12.10.2011	Piotr Janaszek	Added User Interface to Requirements
0.7	14.10.2011	Przemysław Rosiński	Created Main User Paths and Basic Sources For Needed Data
0.9	16.10.2011	Piotr Janaszek	Created Postface
1.0	17.10.2011	Przemysław Rosiński	Fixed mistakes and changed document formatting
1.1	20.10.2011	Przemysław Rosiński	Changed Preface, Requirements and Main User Paths sections

PREFACE FOR MANAGEMENT

The aim of the project is to develop a generic C# library of some search trees algorithms. The library should be standardized in order to be utilizable in other applications.

The developers are also said to provide a sample application showing capabilities of this library. In order to do so the program should be able to retrieve necessary data about the tree manually by user input via GUI (Graphical User Interface) and/or automatically from external file. Data in external file can be from different distributions generated in R package.

The application should graphically present structure of the tree. Result of algorithm execution should be also visualized by marking the path from root to the particular node.

Program must give the user possibility to compare the performance of search trees. It is required that the application allows the user to use R package for the statistical purposes by exporting file compliant with this tool.

Input file as well as output file must be compatible with R. It is agreed deliberately that the application must be able to work with CSV (Comma Separated Values) file format.

REQUIREMENTS

Input data

- Type of the tree can be selected via GUI.
- Type of the data in nodes can be selected via GUI.
- Nodes can be added:
 - automatically from external file,
 - manually via GUI.
- Nodes can be deleted via GUI.

Processing

- Tree creating and searching algorithm can be executed in the background.
- Processing time must be calculated.
- Only one tree can be processed at the time.
- Many trees can be processed sequentially.

Results

- After every addition/deletion of nodes generated tree will be displayed to the user.
- After executing searching algorithm, the result in the form of marked path from root to expected node will be also visualized.
- Time of running the algorithm will be displayed.
- Performance results can be saved into external file.

User interface

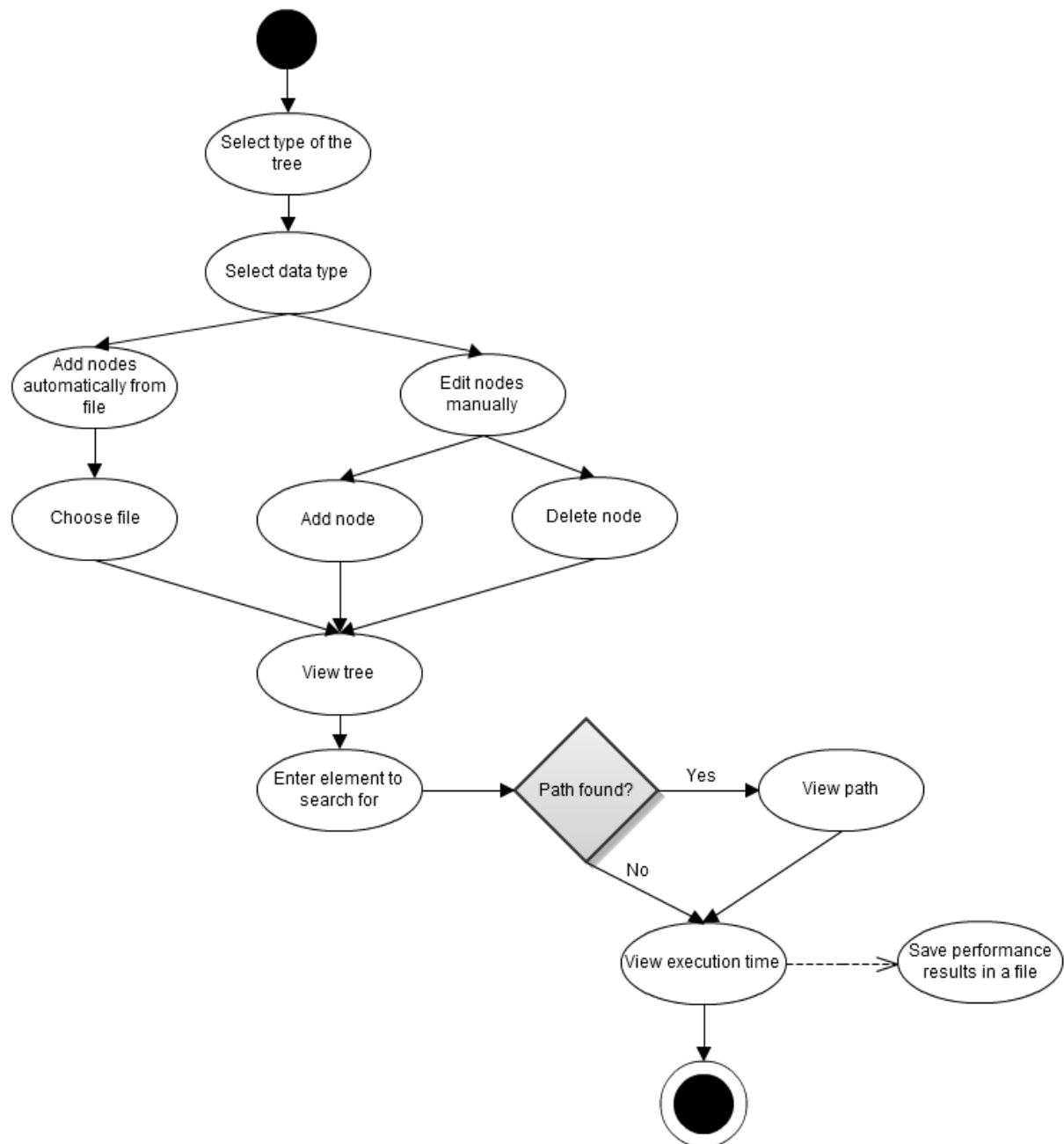
- User interface is graphical.
- The layout is clear and well-formed.
- Buttons are described in words shortly and precisely or by symbols, where they are sufficient.
- Descriptions of buttons and functions are intuitive, easy to understand and not confusing the user.

Data types

- Required search trees are: BST, AVL, red-black, 2-3, 2-3-4, B, B+, Splay.
- Required data types of nodes content are: numerical, text.

MAIN USER PATHS

There is main user path of using the program presented below:



Drawing 1: Main user path diagram. Diagram created using gliffy.com

BASIC SOURCES OF NEEDED DATA

The instance of the problem solved by the application consists of search tree and searched element.

Below, there is presented needed data and its sources:

- Tree type - user input.
- Nodes internal data type - user input.
- Nodes content - external file and/or user input.
- Searched element - user input.

POSTFACE

The application functionality is rather simple and it is designed to execute searching algorithms in different data structures. Program job should allow the user to perform the comparison between received results by storing sufficient information in standardized file.

The application also shows the user the way of storing data in the search trees.