Term Projects in Data Structures

for COS 226: Introduction to Data Structures

Sudarshan S. Chawathe

University of Maine

Fall 2007

A few project ideas that may be developed further into suitable term projects are outlined below. The three representative projects below illustrate different kinds of projects: The first is based on a single, complex algorithm for a relatively simple problem. The second is based on a comparative study of a number of related data structures. The third focuses on an end-user application.

Several variations of these projects are possible. For example, there is a great variety of data structures from which to choose a few for experimental study in the Dynamic Search project.

The following outlines focus on the distinguishing tasks in each project. Tasks that are common to all projects, such as studying and documenting related work, writing robust and readable code, documenting algorithms and code, using proper experimental procedures, and creating suitable reports and presentations, are skipped.

Medians The focus of this project is an optimized implementation and experimental study of the linear-time median algorithm by Schonhage, Paterson, and Pippenger [1]. Some key tasks specific to this project are:

- Reading the paper to thoroughly understand the algorithm, which is fairly complex.
- Deciding how best to implement various constructs used by the algorithm, such as factories for partially ordered elements.
- Implementing the algorithm and validating both its correctness and performance.
- Comparing the implementation's performance with that of other efficient implementations median-finding algorithms, including alternate linear-time algorithms [2] as well as super-linear time algorithms that are based on sorting, quick-selection, and other methods.

Dynamic Search The focus of this project is an experimental comparison of several data structures for dynamic search, such as AVL trees, red-black trees, AA-trees, splay trees, and skip lists. Heger's paper on the topic provides an example of such work [3]. Some key tasks specific to this project are:

- Identifying and documenting available implementations of suitable data structures.
- Modifying available implementations of data structures and implementing others in a manner that facilitates useful comparisons.
- Compiling a comprehensive suite of tests to be used for evaluating the implementations, with attention to covering problematic special cases as well as expected typical cases, based on a few target applications.
- Conducting a thorough experimental study to compare the implementations.
- Explaining the experimental results qualitatively focusing on enduring properties and, in particular, separating implementation artifacts from inherent properties of the data structures.
- Comparing the results with those in prior work and, in particular, explaining any significant differences.

File-System Search The focus of this project is an application: searching for files in a large file-system. Some key tasks specific to this project are:

- Specifying application requirements in detail, with attention to details such as the data that is to be searchable, the query language used for searching, the user interface, and the performance constraints (e.g., response time for queries, running time and load for background indexing, and space used by indexes).
- Identifying existing applications that are similar to the proposed one, with particular attention to identifying similarities and differences.
- Identifying standard data structures used for indexing various data types required by the application and evaluating their effectiveness for this application, using a combination of methods (prior results, experiments, etc.).
- Implementing the complete application using a combination of data structures and related methods.
- Experimentally evaluating the performance of the application on real (or realistic) inputs.

References

[1] A. Schonhage, M. Paterson, and N. Pippenger. Finding the median. *Journal of Computer and System Sciences*, 13:184–199, 1976.

[2] Derrick Coetzee. An efficient implementation of Blum, Floyd, Pratt, Rivest, and Tarjan's worst-case linear selection algorithm. `http://moonflare.com/`, January 2004.

[3] Dominique A. Heger. A disquisition on the performance behavior of binary search tree data structures. *European Journal for the Informatics Professional*, 5(5):67–75, October 2004.