

Class 44: Binary Search

Held: Wednesday, 18 November 2009

Summary: We consider the general problem of searching and *binary search*, one of the most efficient algorithms for searching.

Related Pages:

- EBoard.
- Lab: Binary Search.
- Reading: Search Algorithms.
- Due: Assignment 8: Trees; Project Proposal.

Notes:

- I'm a bit disappointed with the increasing number of people missing class.
- There is no reading for Friday. Use the extra time to work on your projects.
- Project proposals due today.
- Reminder: Projects are due next *Tuesday*.
- EC for Thursday's convocation.
- EC for Thursday's CS Extra on Bioinformatics (Thursday at 4:15).
- EC for Friday's CS Table on "Computational Thinking" (readings available in class today).
- EC for Friday night's swim meet.

Overview:

- The problem of searching.
- Analyzing algorithmic efficiency.
- Making searching more efficient using divide-and-conquer.

Common Problems and Algorithms

- As we discussed early in the semester, a key aspect of computer science is the design of *algorithms*, formalized processes that provide solutions to problems.
- There are a number of common problems for which computer scientists have developed common solutions.
- We'll visit two problems over the next few days: searching and sorting.
- As we develop algorithms, we'll consider intuitive ways that one might come up with the algorithms.

Searching

- Goal: Find a value in a collection.
- Typically, the collection is *linear*: A vector or list.
- Sometimes, the collection is also *unordered*. That is, there is no known arrangement to the list. For example, the books on the MathLan book shelves are not in an arrangement that would make it easy to search for a book with a particular title or by a particular author.
- For unordered collections, the typical search is *sequential search*, look at each element in turn.
- Sometimes, the collection is *sorted*. That is, the collection is organized by the primary key in which we search.
 - For example, a phone book is sorted by name.
- However, we can also use something known as *binary search*:
 - Look in the middle of the collection.
 - If the middle is too small, anything smaller is also too small, so discard and try again.
 - If the middle is too large, anything larger is also too large, so discard and try again.
 - If the middle is just right, you're done.

Lab

- Do The lab.
- Be prepared to reflect.

Copyright © 2007-9 Janet Davis, Matthew Kluber, Samuel A. Rebelsky, and Jerod Weinman. (Selected materials copyright by John David Stone and Henry Walker and used by permission.) This material is based upon work partially supported by the National Science Foundation under Grant No. CCLI-0633090. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This work is licensed under a Creative Commons Attribution-NonCommercial 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.