Programming Report # 3

Searching

Due March 1, 2010

For this report, you are going to test two different searching algorithms, sequential and binary. As with all experimental reports, you are welcome to use any language you wish and implement the required programming aspects however you choose, but all language decisions as well as implementation details must be specified in the report.

Sequential Search v. Binary Search

Sequential search is known to be slower than binary search for large values of N. Given a particular computing environment (OS, hardware, etc.), is there a value of N for which binary search provides little to no improvement over sequential search? Given those environmental constraints, where is the break-point between the two (i.e. for what value of N does binary search out-perform sequential consistently), if any? How does the hit/miss ratio affect performance of both?

You will need to implement both binary and sequential search. You will need to test them against each other for various values of N, on the **same** non-decreasing data sets (strictly increasing sets are fine) and on the **same** query sets. You will need also to vary the hit-miss ratio of your queries to determine the significance of its affect, if any. You will need to test this framework on at least **two** different hardware setups (e.g. the one in the lab and one of the vulcan machines in the undergraduate lab).

All answers to the above questions need visual demonstration.

Discuss these issues:

- Is sequential search slower than binary search?
- If there is a break point, why does it exist?
- How does hit/miss ratio of queries affect the two algorithms' performance?
- Discuss some potential reasons for differences in results between computing environments, if any.
- Based on your results, discuss the relative value of sequential and binary search algorithms in terms of search system design.