Ian Buitrago
John Richter
4-30-2012

CS 337 project 3: string matching Report

First we compare the brute force algorithm to the other 3 algorithms, Rabin-Karp, Kunth-Morris-Pratt, and Boyer-Moore. RK performs better than BF. BM is the fastest algorithm.

Case 1: “Richard Dawkins” in Bible

In our first case, we wanted to determine the speed at which our algorithms could go through the entire bible looking for an obviously nonexistent string, Richard Dawkins. As expected, BF took the longest, while the following three were successively faster. BM performed the best, at 0.25 s.

As far as number of comparisons, Brute force had the most comparisons, followed by KMP, BM, and finally RK. RK had the fewest because it only does character comparisons if the hash is the same.

BF	0.056 sec
RK	0.049 sec
KMP	0.09 sec
BM	0.025 sec

Times

Number of Comparisons

BF	4459694
RK	8665
KMP	4452069
BM	430589

Case 1b:

Now we will compare 2 different hash algorithms. hash() is a linear sum of the ascii values of the characters in the alignment or pattern. hashBase() treats the string as a base 246 number with each term modded by a prime, 31, 127, or 518.

Then we will compare the collisions with the different mod values.

Appendix A: Test Cases

(a) Bible Pattern

(b) Simple Pattern

(c)