CHRISTOPHER FOLEY
Z15092976
PROJECT PROPOSAL

03/21/17

OVERVIEW

1. Project Background and Description

Substring matching is a critical part of modern design. Documents and text readers need to be searched for specific values. Network devices deliver their results in a text based format.

2. Project Scope

This project will investigate algorithms for substring matching in string buffers of varying size. Contents of the string buffer will be restricted to random printable ASCII characters. Search strings will be created and average search time calculated. To force worst case scenario, the search string will contain a non printing character which will not match the buffer. Tests will be performed numerous times on each generated buffer and the worst case search time and buffer size will be written to standard output.

The goal of the project will be to compare substring matching times of two and possibly three algorithms:

- naive matching
- Rabin-Karp
- Knuth-Morris-Pratt

Data will be collected concerning pre processing and string matching times for randomly determined matching and non matching strings. The buffer size will be the independent variable and the search times will be compared.

3. High-Level Requirements

The new system must include the following:

- All test code will be written in C/C++.
- No system library modules will be used to compare strings or substrings during the test phase.
- A spreadsheet program similar to Libre Office Calc will be used for data analysis and chart generation.

4. Deliverables

The project deliverables will be a project report containing pseudocode of the algoithms used with analysis of the output and graphs showing the relationship between the search string buffer size (independent variable) and the time to search. All calculations and analysis will be in the report. Raw data, if not excessive (> 2 pages) will be included as an appendix.

5. Specific Exclusions from Scope

Performance of system libraries provided with the gcc compiler and linker will not be measured.

6. Implementation Plan

The naive buffer will be tested first, followed by Rabin-Karp, followed by Knuth-Morris-Pratt. Each test will be run at least 8 times to ensure sufficient timings are collected. It is intended that the test mechanism be command line driven with the ability to operate unattended. Text output will be collected in a data file for analysis.

1. String buffer sizes

The string source buffer sizes will begin at 1GB and decrease to 1024 bytes decrease in powers of 2, if the timings of individual runs indicate that the tests can not be completed in a reasonable amount of time. A base buffer of the maximum size will be created with random printable ASCII characters except the '@' character. The string source buffer will not be altered during the test.

The Rabin-Karp algorithm requires the use of an additional buffer to store numeric values from preprocessing. If the string and alternate buffers can not be created, the sizes will be reduced at the start of the test.

String search buffers will be reflective of "real world" and be in sizes of the form 2^n-1 with n=1..8. Since a worst case scenario is desired, buffer sizes having odd parity and containing the the '@' as the last character will be used to guarantee no matching. The search string will be reinitialized prior to each test.

The search string buffer and temporary buffers created will be returned to the system at the conclusion of each test. The source buffer will be returned at the conclusion of all tests.

2. Data Presentation

Data will be presented in tabular and chart form with the search buffer size as the independent variable and the search time as dependent.

Other tables will be added to show estimates of values used to calculate algorithm limits and efficiency.

Tables and charts containing the theoretical and experimental models for each algorithm will be created. A final summary chart comparing the worst case performance of all values will be created.

3. Test Execution

Test modules will be developed as command line driven and where possible, script driven. Output will be captured to a data file which will be used to provide input to the final results.

4. Worst Case Runtime

When determining the Worst Case scenario runtime, there are two considerations. The search buffer (n) and the pattern buffer(m) must be taken into account. Since the search buffer (n) is of the order 10^9 and the pattern buffer (m) is 1024 bytes all algorithms will operate on the order of 10^{18} for the Naive matching (with an anticipated time of 10^{18} / CPU time) . Others will operate on a more linearly related time frame closer to the search buffer size / CPU time.

A review of “Introduction To Algorithms” (2nd Edition) indicates that worst case times may be of the following orders:

Algorithm	Preprocessing	Matching Time
Naive	0	*O((n-m+1)m)*
Rabin-Karp	θ(m)	*O((n-m+1)m)*
Knuth-Morris-Pratt (KMP)	θ(m)	*θ(n)*

5. Other Potential Considerations

- In addition to the primary test machine, a 166Mhz Pentium-II CPU based Linux machine may be used to collect additional data.
- During development it may be determined that worse case times may be determined by alteration of the contents of the buffer, the buffers for search and pattern matching may be altered as needed. It anticipated that a worst case matching will require that the search buffer be initialized to one character or character sequence and the pattern buffer match a portion of the buffer.
- Possible buffer variations should occur due to the algorithm used. If different search/pattern buffer contents are used, the new patterns will be used on all algorithms.
    - Naive worst case should be a 2 character pattern buffer with the final character not matching.
    - Rabin-Karp converts a range of characters to a numeric value numeric values which adjust the parity may be of concern.
    - KMP uses shifts and small mismatches are worst case.

7. High-Level Time line/Schedule

Coding will begin as soon as practical after approval and all deliverables delivered by 11:59Pm on April 21, 2017. It is anticipated that the bulk of the work will begin after April 2, 2017 with a delay during April 7-9, 2017.

APPROVAL AND AUTHORITY TO PROCEED

Signature authority is not required. Approval may be given by email.

We approve the project as described above, and authorize the team to proceed.

Name	Title	Date

Approved By	Date	Approved By	Date