Ottocento Tommaso 202002457
Hansen Oliver 201708342
Pediotidis Maniatis Dimitris 202002459 September 2021

Genome Scale Algorithms - Project 1

Introduction

In this project we implemented two algorithms for exact pattern matching, the naive quadratic algorithm and the linear border-array. We performed experiments to verify the correctness of both algorithms and also their running times which can be seen in the included plots. Each algorithm takes two inputs, a FASTA and a FASTQ file, and outputs all matches in SAM format to stdout.

Problems.

Our first implementation of the border-array matching was seemingly working correctly, but after trying various inputs we noticed that due to a mistake in indexing it actually did not go in the final loop of the algorithm and thus was unable to find the longest border of the input string at the last index. This was addressed and fixed. We cannot explain the vast difference in the running time of the Worst Case Border Array as can be seen in figure 4

Verification of Correctness.

We used known pattern occurrences in strings calculated by hand to verify that our program calculates them correctly. Additionally we used the Mississippi example.

Experiments

We tested both algorithms for input string length varying from 10000 to 500000 and pattern length as 2 , 100 and 1000. We tested for randomly generated strings and the worst case scenario : Input string $x = a^n$ and pattern $p = a^m$, $m < n \in \mathbb{N}$.

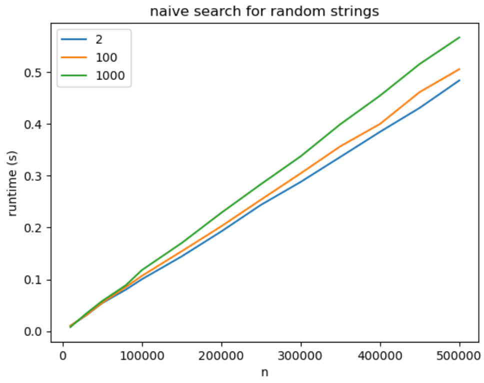

Figure 1: Naive Random Strings Search Times

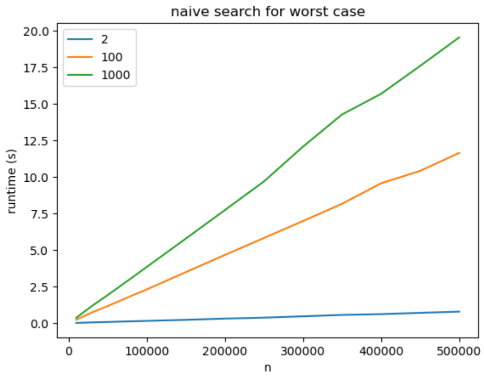

Figure 2: Naive Worst cases Search Times

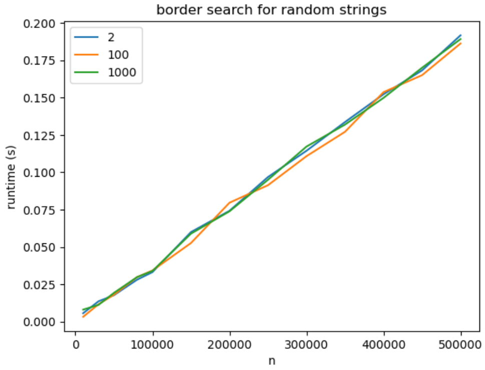

Figure 3: Border Search Random Strings Search Times

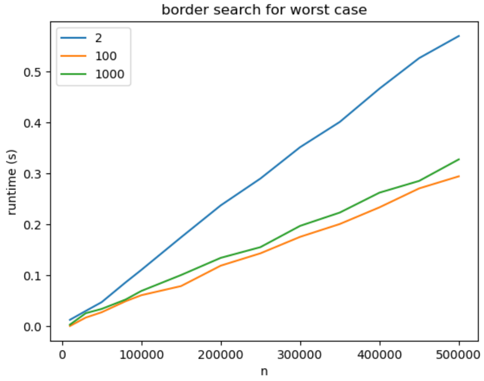

Figure 4: Border Search Worst Case Running Times

Remarks

As expected we see a considerable difference in running time between the algorithms. While there are both bounded by n in i time as they appear linear we can see the naive search uses more time because its $\mathcal{O}(n * m)$. The difference between the random and worst is significant and we can see how the naive search running is affected more by the increase in m, then the border search is.