Practical 8_Searching Algorithms

Brute Force Algorithm - performance

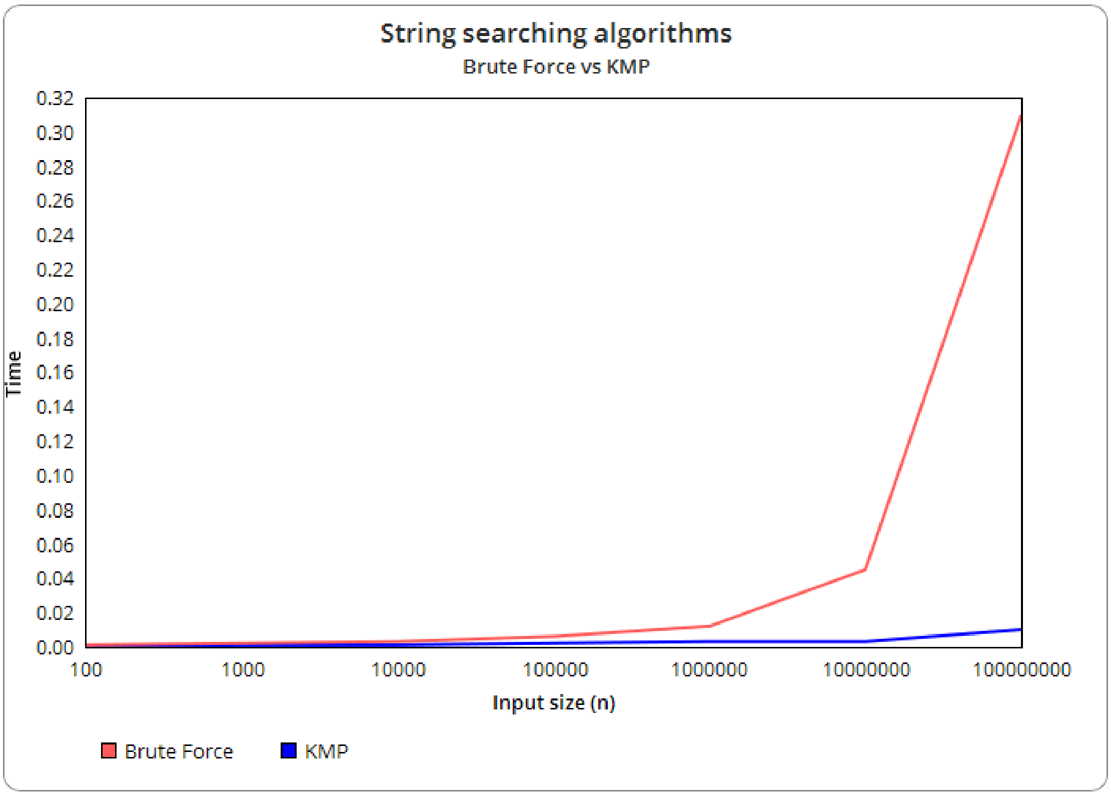
This algorithm works by testing all the possible placement of a given pattern within a string of text. It consists of two nested loops, the outer loop spans from 0 to the end of the text length (n), while the inner loop spans from 0 to the end of the pattern length (m). For this reason, the time complexity of the brute force algorithm is O(m*n). It is important to note that depending on the length of m, the algorithm could potentially have a worst case performance of $O(n^2)$, which is not very efficient.

Knuth-Morris-Pratt Algorithm - performance

The main idea of the KMP algorithm is to reuse previously performed comparisons. By doing this, the algorithm achieves a worst case performance of O(n+m) - meaning that it will have to examine the entire text and the entire pattern at most once - which is a big improvement in efficiency compared to the brute force algorithm.

String Searching Algorithms - Analysis

Input size (n)	Time (Brute Force)	Time (KMP)
100	0.001	0.0
1000	0.002	0.001
10000	0.003	0.001
100000	0.006	0.002
1000000	0.012	0.003
10000000	0.045	0.003
100000000	0.310	0.010


String searching algorithms
Brute Force vs KMP
Time
Input size (n)
🟥 Brute Force
🟦 KMP