Algorithm programming assignment – 2

# Shyam Kantesariya(40042715)

Linear search has complexity n2 where in using SuffixTree we can improve it to O(n) in worst case, considering we have O(1) complexity to search for any alphabet in the tree.

Following are the various runtimes observed for different combinations of Text and Pattern Length.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TC# | Text Length | Pattern Length | Linear | SuffixTree |
| 1 | 40000 | 50 | 0 | 0 |
| 2 | 40000 | 1000 | 0 | 0 |
| 3 | 40000 | 5000 | 0 | 0 |
| 4 | 40000 | 9000 | 0 | 0 |
| 5 | 40000 | 10000 | 0 | 0 |
| 6 | 100000 | 50 | 0 | 0 |
| 7 | 100000 | 1000 | 0 | 0 |
| 8 | 100000 | 5000 | 0 | 0 |
| 9 | 100000 | 9000 | 0 | 0 |
| 10 | 100000 | 10000 | 0 | 0 |
| 11 | 250000 | 50 | 0.04 | 0.05 |
| 12 | 250000 | 1000 | 0.07 | 0.06 |
| 13 | 250000 | 5000 | 1.3 | 0.08 |
| 14 | 250000 | 9000 | 1.9 | 1.07 |
| 15 | 250000 | 10000 | 2.5 | 1.2 |
| 16 | 400000 | 50 | 0.12 | 0.08 |
| 17 | 400000 | 1000 | 0.15 | 0.09 |
| 18 | 400000 | 5000 | 1.7 | 1.03 |
| 19 | 400000 | 9000 | 3.06 | 1.2 |
| 20 | 400000 | 10000 | 4.18 | 1.8 |

From the observation, we can see that runtime doesn’t give much difference for small text and pattern length. But as we increase the length of these two, we see a considerable difference in runtime.

SuffixTree works better compare to linear search as length of text/pattern increases.

Following are the details for code:

|  |  |  |
| --- | --- | --- |
| **Algorithm type** | **main class name** | **Remark** |
| Linear | LinearPatternMatcher |  |
| Suffix Tree with Node array | SuffixTreePatternMatcher | Takes lot of time to build tree with large text |
| Suffix Tree with Node and Edges | SuffixTreePatternMatcherWithEdges | Referred GeekForGeeks to develop this efficient DS. Used hash map instead of Array |