

## Relatório

Experimentalmente, verifiquei que o maior valor de N para o qual consegui calcular o L, em meu computador core i5, foi na ordem de 10 milhões. Para N = 100 milhões a memória do meu computador (4 gb) não foi o suficiente.

Analisando as tabelas 1 e 2, é possível inferir que a melhor alternativa para a resolução desse tipo de problema em termos de tempo é a 10-way trie, pois ela possui o melhor desempenho na maioria dos testes e não deu overhead de memória em todos os testes. A pior alternativa, nesse caso, seria utilizar a tabela de hash que consome mais memória e não apresentou melhor resultado em nenhum dos casos.

Para um arquivo pi de 1 milhão de dígitos o programa encontrou um L máximo de 4, para o de 10 milhões foi encontrado um L máximo de 5. Ambos executaram rapidamente (1.882 s e 28.642) na TST.

Tabela 1 (pi com 1 milhão e 10 milhões de dígitos)

| Tabela de símbolo/arquivo | Pi 1 milhão (máximo L = 4) | Pi 10 milhões (máximo L = 5) |
|---------------------------|----------------------------|------------------------------|
| TST                       | 1.896 s                    | 28.642 s                     |
| RedBlackBST               | 2.819 s                    | 55.412 s                     |
| SeparateChainingHashST    | 2.171 s                    | Overhead de memória          |
| 10-way trie               | 1.223 s                    | 32.513 s                     |

Tabela 2 (N dígitos pseudo aleatórios)

| Tabela de símbolo / N  | 10 <sup>4</sup> | 10 <sup>5</sup> | 10 <sup>6</sup> | 10 <sup>7</sup> |
|------------------------|-----------------|-----------------|-----------------|-----------------|
| TST                    | 0.094 s         | 0.247 s         | 1.848 s         | 32.187 s        |
| RedBlackBST            | 0.107 s         | 0.271 s         | 2.9 s           | 55.903 s        |
| SeparateChainingHashST | 0.08 s          | 0.243 s         | 2.274 s         | 50.448 s        |
| 10-way trie            | 0.062 s         | 0.14 s          | 0.933 s         | 29.162 s        |