|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | TinyTale.txt | | Tale.txt | | leipzig1M.txt | |
| Words | Distinct | words | distinct | words | distinct |
| 3 caracteres | 40 | 18 | 106181 | 10608 | 17466737 | 533214 |
| 6 caracteres | 10 | 9 | 33898 | 8395 | 8178792 | 445844 |
| 9 caracteres | 2 | 2 | 8304 | 3557 | 2728360 | 227040 |
| 12 caracteres | 0 | 0 | 1250 | 754 | 493299 | 79900 |

**Tiempo de ejecución en segundo con el documento TinyTale.txt.**

|  |  |  |  |
| --- | --- | --- | --- |
| TinyTale.txt | | | |
|  | BUSQUEDA BINARIA | ARBOL ROJO NEGRO | TABLA HASH |
| 3 caracteres | 0 | 0 | 0 |
| 6 caracteres | 0.01 | 0 | 0 |
| 9 caracteres | 0 | 0 | 0 |
| 12 caracteres | 0 | 0 | 0 |

**Tiempo de ejecución en segundo con el documento Tale.txt**

|  |  |  |  |
| --- | --- | --- | --- |
| Tale.txt | | | |
|  | BUSQUEDA BINARIA | ARBOL ROJO NEGRO | TABLA HASH |
| 3 caracteres | 0.355 | 0.149 | 0.038 |
| 6 caracteres | 0.137 | 0.061 | 0.017 |
| 9 caracteres | 0.036 | 0.016 | 0.016 |
| 12 caracteres | 0.009 | 0.008 | 0.002 |

**Tiempo de ejecución en segundo con el documento leipzig1M.txt**

|  |  |  |  |
| --- | --- | --- | --- |
| Leipzig1M.txt | | | |
|  | BUSQUEDA BINARIA | ARBOL ROJO NEGRO | TABLA HASH |
| 3 caracteres | 180.756 | 5.678 | 1.004 |
| 6 caracteres | 93.568 | 2.568 | 0.756 |
| 9 caracteres | 87.245 | 1.958 | 0.439 |
| 12 caracteres | 80.655 | 1.435 | 0.373 |

**Conclusión**

He llegado a conclusión que el algoritmo Tabla Hash con Consulta Lineal es más rápido que los demás algoritmos en búsqueda, sin importar el tamaño del documento o el tamaño de los caracteres el algoritmo más rápido es el Tabla Hash con Consulta Lineal luego el Árboles Rojo-Negro y finalmente el de Búsqueda Binaria.