**Computador:** Notebook CCE Aspire E 15 E5-574g-75ME

**Sistema Operacional**: Linux Ubuntu 16.04

**Processador**: Intel Core i7, 6° geração

**Ram**: 8 Gb

**Dupla**: Francisco Ferreira da Silva Junior (377588) e Maria Juliana Pontes Mateus (373703)

**Resultados Totais**

|  |
| --- |
| HEAP |
|  | OperaçõesA | OperaçõesI | OperaçõesR | OperaçõesS |
| 100 | 10 | 8 | 7 | 8 |
| 10000 | 190 | 104 | 115 | 140 |
| 50000 | 409 | 471 | 443 | 359 |
| 100000 | 815 | 868 | 752 | 841 |
| 500000 | 3773 | 3546 | 3822 | 3951 |
| 800000 | 7345 | 6732 | 6305 | 6132 |

|  |
| --- |
| Não ordenado |
|  | OperaçõesA | OperaçõesI | OperaçõesR | OperaçõesS |
| 100 | 8 | 7 | 9 | 9 |
| 10000 | 170 | 182 | 122 | 152 |
| 50000 | 406 | 392 | 430 | 397 |
| 100000 | 870 | 833 | 704 | 868 |
| 500000 | 3867 | 3511 | 3658 | 3709 |
| 800000 | 7287 | 6853 | 5630 | 7068 |

|  |
| --- |
| Ordenado |
|  | OperaçõesA | OperaçõesI | OperaçõesR | OperaçõesS |
| 100 | 9 | 7 | 10 | 8 |
| 10000 | 164 | 141 | 160 | 162 |
| 50000 | 453 | 433 | 486 | 502 |
| 100000 | 970 | 852 | 852 | 848 |
| 500000 | 4170 | 4028 | 3823 | 4054 |
| 800000 | 6625 | 6303 | 6545 | 6363 |

­

**Total de tempo por operação**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | OperaçõesA | OperaçõesI | OperaçõesR | OperaçõesS |
| Heap | 2090 | 1954 | 1050 | 1905 |
| Ordenado | 2101 | 1963 | 1740 | 2033 |
| Não Ordenado | 2065 | 1960 | 1758 | 1989 |

De acordo com os dados vistos, a operação de Alteração requer mais tempo para ser executado. Já a operação de remover requer menos tempo.

Nos arquivos que possuem maior número de operação de alteração a Lista de Prioridade Não Ordenado se mostrou mais rápido.

Nos arquivos que possuem maior número de operação de Inserção o Heap se mostrou mais rápido.

Nos arquivos que possuem maior número de operação de Remoção o Heap se mostrou mais rápido. Para os tipos de entrada dos arquivos lidos, foi observado que o Heap é 40% mais rápido que o Não Ordenado, pois no processo do seu algoritmo o número de maior prioridade está no início do vetor, sendo assim mais fácil de encontrar do que em um vetor não ordenado.

Nos arquivos que possuem maior número de operação de Seleção o Heap se mostrou mais rápido.

Vimos então que, a Lista de Prioridade Ordenada é a mais lenta dos três em todas as Operações. Enquanto o Heap se mostra rápido nas operações em questão.