Análise Lista de Prioridade  
Estrutura de Dados Avançado

**Responsáveis:** João Victor Venancio Santiago, Maria Beatriz de Almeida Batista  
**Professor:** Fábio

UFC – Campus Quixadá – 2017.1

**Informações sobre o Projeto:**

Estruturas utilizadas para estudo:

* Vetor em HEAP
* Lista Não Ordenada
* Lista Ordenada – QuickSort

Ambiente de Desenvolvimento Integrado (Integrated Development Environment – IDE)

* Eclipse Mars.2

**Sobre a máquina utilizada:**

NoteBook: Positivo Master  
Processador: Intel® Core™ i3 – 3110M CPU @ 2.40 GHz  
Memória Instalada(RAM): 5,00 GB  
Tipo de Sistema: Windows 10 x64

**Analisando Estruturas de Prioridade**

Legenda:

**Operações\_I:** Tempo de operação para inserção  
**Operações\_A:** Tempo de operação para alteração  
**Operações\_R:** Tempo de operação para remoção  
**Operações\_S:** Tempo de operação para seleção

Vetor em HEAP Máximo

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 100 | 10000 | 50000 | 100000 | 500000 | 800000 |
| Operações\_I | 14 | 303 | 1181 | 2422 | 13887 | 22584 |
| Operações\_A | 8 | 182 | 1257 | 2583 | 15638 | 24654 |
| Operações\_R | 7 | 107 | 1188 | 2860 | 13968 | 23936 |
| Operações\_S | 3 | 189 | 1156 | 2545 | 13192 | 21634 |

Lista de Prioridade Não Ordenada

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 100 | 10000 | 50000 | 100000 | 500000 | 800000 |
| Operações\_I | 15 | 410 | 1223 | 2483 | 13844 | 21456 |
| Operações\_A | 9 | 174 | 1265 | 2746 | 16841 | 24432 |
| Operações\_R | 10 | 185 | 1205 | 2815 | 13832 | 24044 |
| Operações\_S | 4 | 169 | 1209 | 2379 | 13909 | 21772 |

Lista de Prioridade Não Ordenada

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 100 | 10000 | 50000 | 100000 | 500000 | 800000 |
| Operações\_I | 39 | 631 | 1562 | 2977 | 25638 | 32245 |
| Operações\_A | 18 | 378 | 1653 | 3065 | 17418 | 30688 |
| Operações\_R | 24 | 395 | 2355 | 3648 | 17956 | 39883 |
| Operações\_S | 8 | 312 | 1495 | 3030 | 16055 | 24966 |

**Conclusão**

A partir da análise das tabelas e do gráfico abaixo, somando o tempo por operação, é possível afirmar a vantagem que Heap oferece sendo utilizado em operações que usam prioridade como base. A Lista Não Ordenada tem desempenho semelhante ao do Heap na operação de remoção, superando o segundo em Inserção.  
 A Lista Ordenada apresenta um comportamento estranho, considerando sua análise de tempo computacional. Por isso, o estado da máquina no período de execução do algoritmo deve ser considerado, assim reforçando a ideia de que o melhor algoritmo só pode ser definido com analise matemática.

O gráfico apresenta o   
tempo total para cada  
operação em seu respectivo  
algoritmo, em prol de facilitar  
a análise do crescimento de  
tempo das operações.