

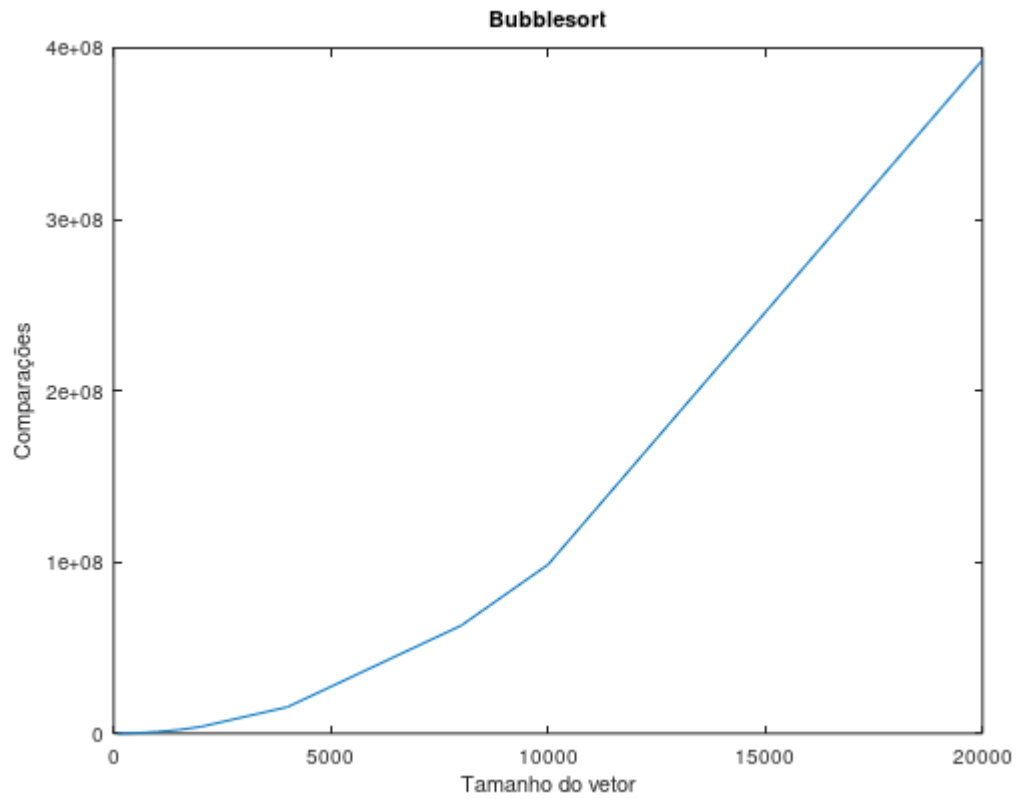
Relatório EP3

O intuito desse ep é analisar como se comportam os algoritmos de ordenação. E a partir disso compará-los e entender quais são suas melhores utilizações. Olharemos a quantidade de comparações e movimentações que cada um deles faz. Testaremos 5 algoritmos: Bubblesort, Insertionsort, Mergesort, Quicksort e Heapsort. Cada um deles terá 4 tipos de testes, cujos vetores serão: randomicos, ordenados crescentemente, ordenados decrescentemente, metade randomicos e metade ordenado crescentemente. Veremos seus comportamentos específicos e cruzaremos informações para concluir alguns pontos de convergência e divergência entre eles. Os gráficos apresentados foram todos gerados pelo Octave.

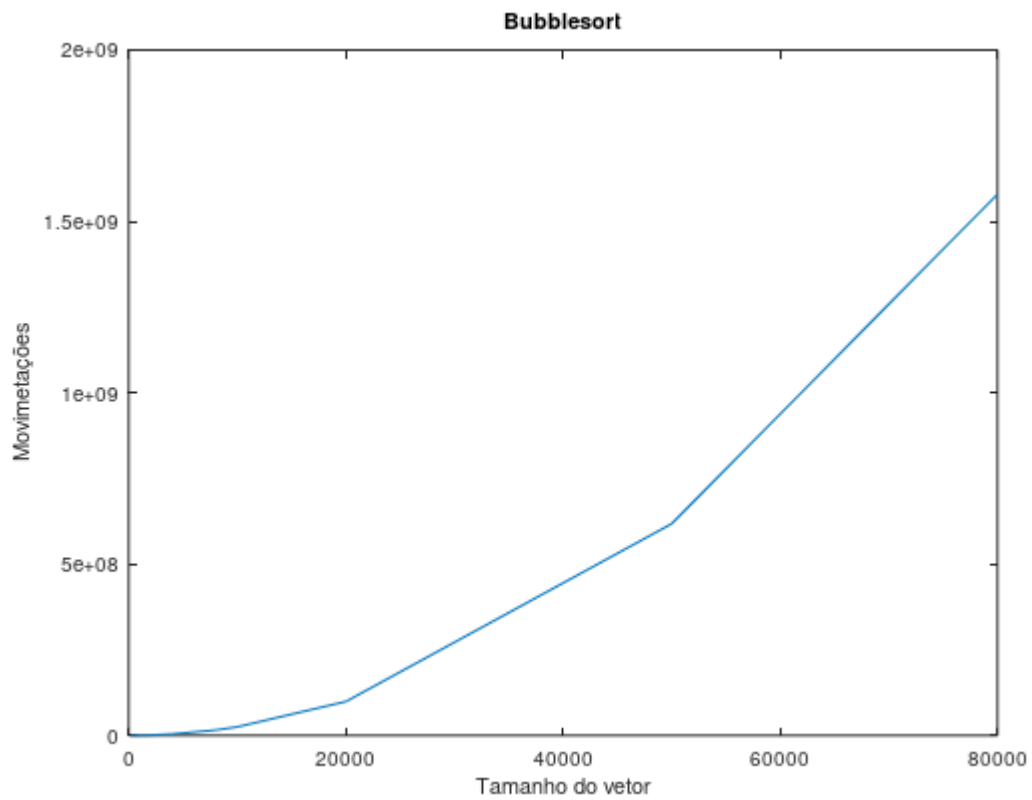
1) BubbleSort

1.1) Número de comparações e movimentações para um vetor totalmente randômico:

Para o vetor de tamanho 250, temos 57270 comparacoes e 16368 movimentacoes
Para o vetor de tamanho 500, temos 235528 comparacoes e 61326 movimentacoes
Para o vetor de tamanho 750, temos 531041 comparacoes e 140275 movimentacoes
Para o vetor de tamanho 1000, temos 928071 comparacoes e 250176 movimentacoes
Para o vetor de tamanho 1250, temos 1513788 comparacoes e 381508 movimentacoes
Para o vetor de tamanho 1500, temos 2176548 comparacoes e 556229 movimentacoes
Para o vetor de tamanho 2000, temos 3882058 comparacoes e 1001869 movimentacoes
Para o vetor de tamanho 4000, temos 15648087 comparacoes e 3963770 movimentacoes
Para o vetor de tamanho 8000, temos 63440069 comparacoes e 15670369 movimentacoes
Para o vetor de tamanho 10000, temos 98100189 comparacoes e 24891535 movimentacoes
Para o vetor de tamanho 20000, temos 395000249 comparacoes e 98940123 movimentacoes
A partir do tamanho 50000 estoura o número de comparações.
Para o vetor de tamanho 50000, temos -1821466767 comparacoes e 617752107 movimentacoes
Para o vetor de tamanho 80000, temos 2040233513 comparacoes e 1579270299 movimentacoes
A partir do tamanho 100000 estoura o número de movimentações.
Para o vetor de tamanho 100000, temos 1309166416 comparacoes e -1814384438 movimentacoes



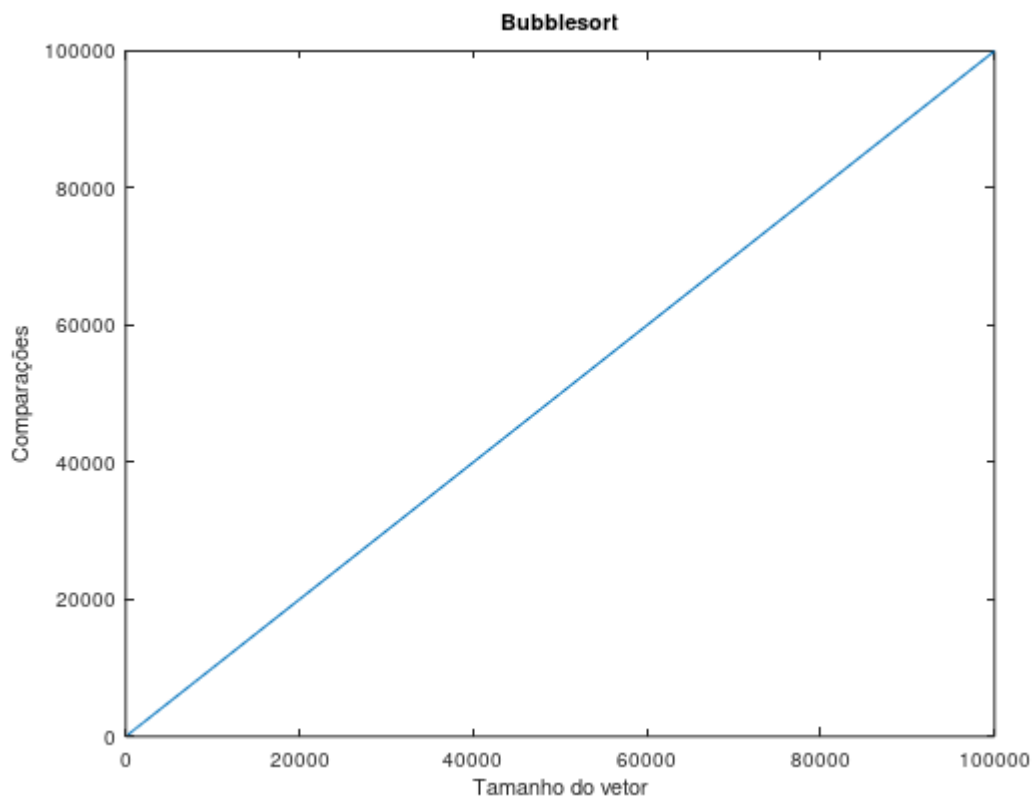
Como esperado, sendo um algoritmo $O(n^2)$, o gráfico de comparações mostra exatamente uma função exponencial.



O número de movimentações também segue o exponencial, só é mais devagar que a quantidade de comparações.

1.2) Número de comparações e movimentações para um vetor crescente:

Para o vetor de tamanho 250, temos 249 comparações e 0 movimentações
Para o vetor de tamanho 500, temos 499 comparações e 0 movimentações
Para o vetor de tamanho 750, temos 749 comparações e 0 movimentações
Para o vetor de tamanho 1000, temos 999 comparações e 0 movimentações
Para o vetor de tamanho 1250, temos 1249 comparações e 0 movimentações
Para o vetor de tamanho 1500, temos 1499 comparações e 0 movimentações
Para o vetor de tamanho 2000, temos 1999 comparações e 0 movimentações
Para o vetor de tamanho 4000, temos 3999 comparações e 0 movimentações
Para o vetor de tamanho 8000, temos 7999 comparações e 0 movimentações
Para o vetor de tamanho 10000, temos 9999 comparações e 0 movimentações
Para o vetor de tamanho 20000, temos 19999 comparações e 0 movimentações
Para o vetor de tamanho 50000, temos 49999 comparações e 0 movimentações
Para o vetor de tamanho 80000, temos 79999 comparações e 0 movimentações
Para o vetor de tamanho 100000, temos 99999 comparações e 0 movimentações

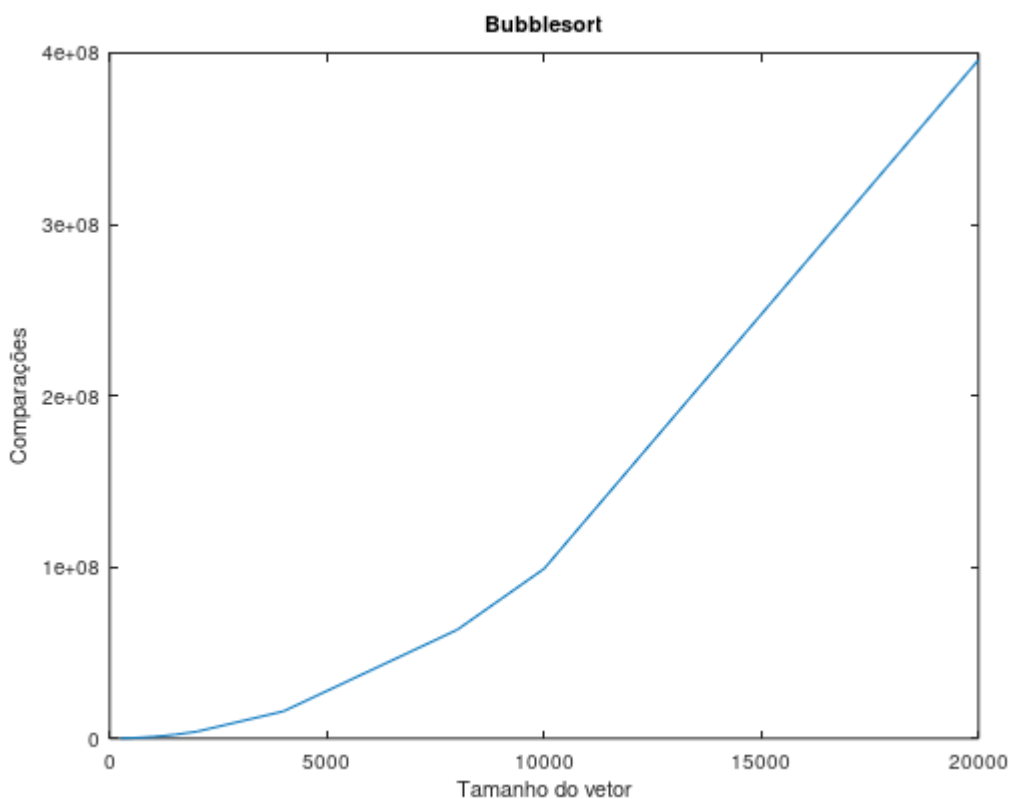


No melhor caso, onde o vetor já está ordenado ele tem $O(n)$ comparações. E zero movimentações.

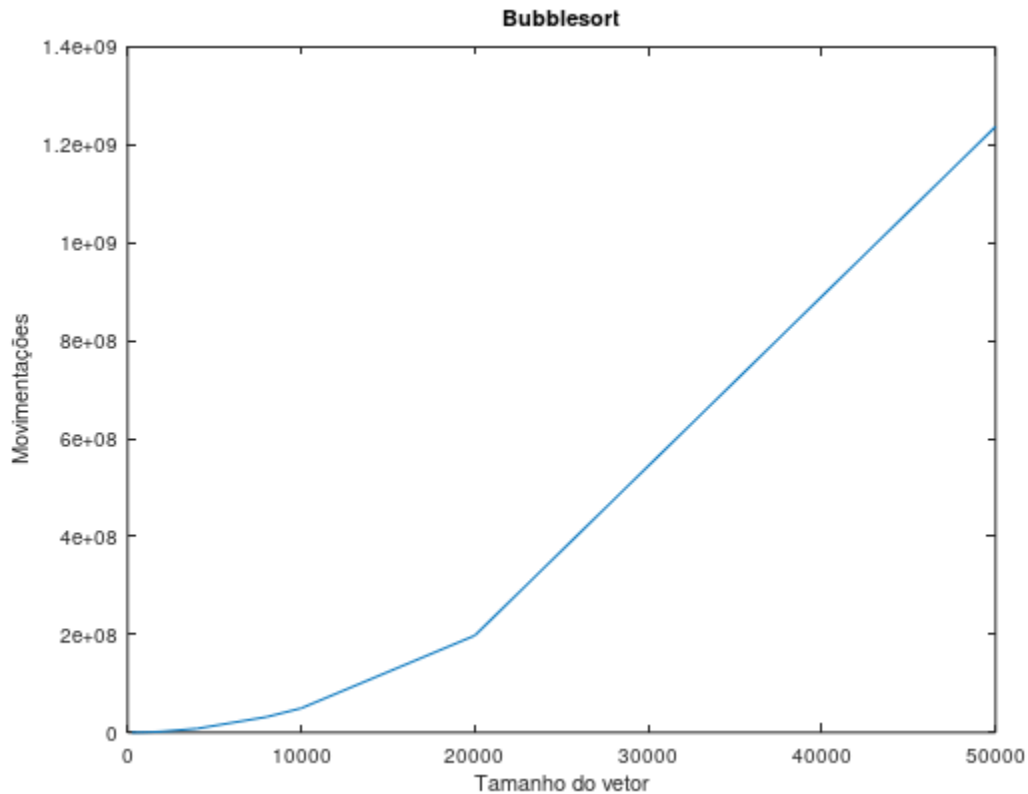
1.3) Número de comparações e movimentações para um vetor decrescente:

Para o vetor de tamanho 250, temos 62001 comparações e 30833 movimentações
Para o vetor de tamanho 500, temos 246506 comparações e 123501 movimentações

Para o vetor de tamanho 750, temos 557256 comparações e 278140 movimentações
Para o vetor de tamanho 1000, temos 992007 comparações e 494609 movimentações
Para o vetor de tamanho 1250, temos 1551258 comparações e 772732 movimentações
Para o vetor de tamanho 1500, temos 2229013 comparações e 1113225 movimentações
Para o vetor de tamanho 2000, temos 3966016 comparações e 1978802 movimentações
Para o vetor de tamanho 4000, temos 15852036 comparações e 7918365 movimentações
Para o vetor de tamanho 8000, temos 63432070 comparações e 31676438 movimentações
Para o vetor de tamanho 10000, temos 99050094 comparações e 49494213 movimentações
Para o vetor de tamanho 20000, temos 396000199 comparações e 197990577 movimentações
A partir do tamanho 50000 estoura o número de comparações.
Para o vetor de tamanho 50000, temos -1819116814 comparações e 1237475225 movimentações
A partir do tamanho 80000 estoura o número de movimentações.
Para o vetor de tamanho 80000, temos 2041113502 comparações e -1127009599 movimentações
Para o vetor de tamanho 100000, temos 1309566412 comparações e 654988013 movimentações



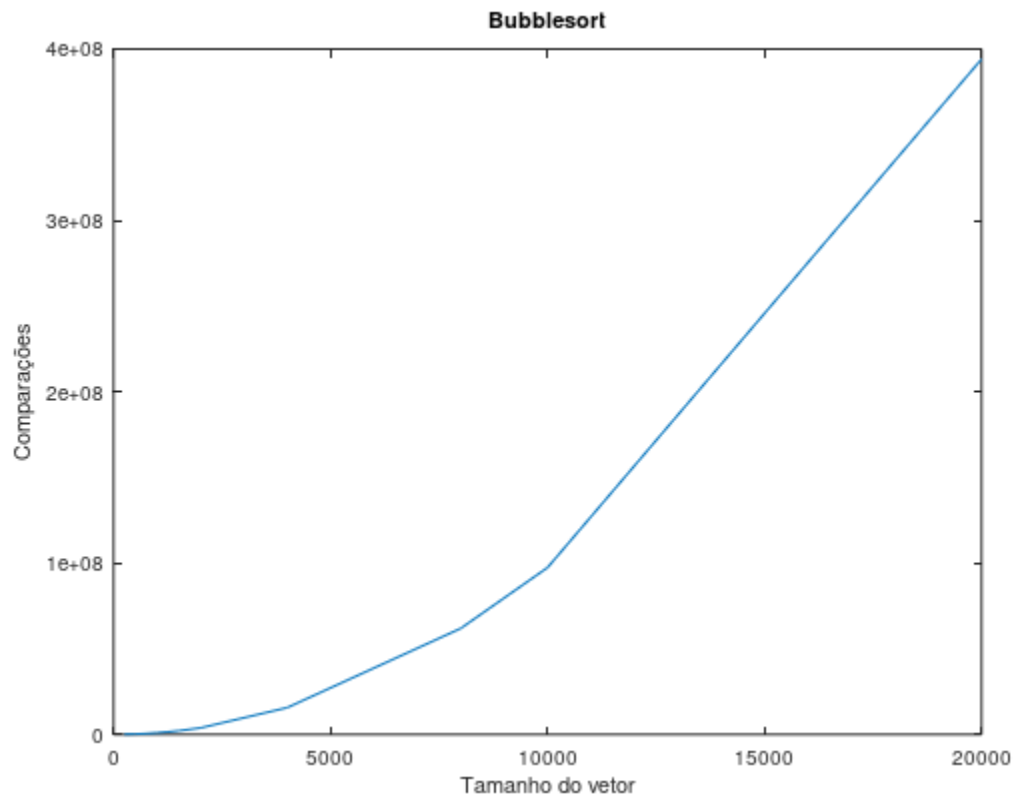
Esse é o pior caso, e fica bem próximo do caso randômico com as comparações. $O(n^2)$ com função exponencial.



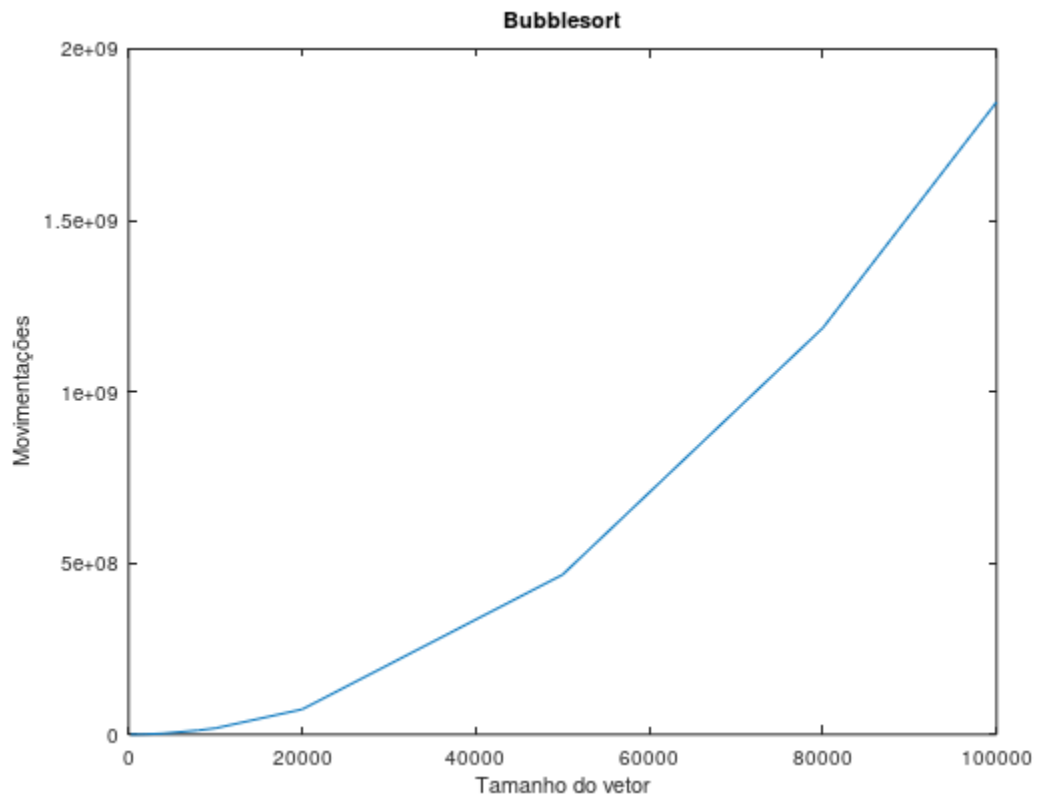
Como se trata do pior caso, o número de movimentações é o máximo possível. Assim tem uma função exponencial mais rápida do que o caso randômico. Como esperado $O(n^2)$.

1.4) Número de comparações e movimentações para um vetor metade ordenado crescentemente e metade randômico:

Para o vetor de tamanho 250, temos 59511 comparações e 11765 movimentações
 Para o vetor de tamanho 500, temos 233033 comparações e 43523 movimentações
 Para o vetor de tamanho 750, temos 528794 comparações e 108574 movimentações
 Para o vetor de tamanho 1000, temos 970029 comparações e 189004 movimentações
 Para o vetor de tamanho 1250, temos 1508792 comparações e 289334 movimentações
 Para o vetor de tamanho 1500, temos 2152564 comparações e 405504 movimentações
 Para o vetor de tamanho 2000, temos 3810094 comparações e 737661 movimentações
 Para o vetor de tamanho 4000, temos 15652086 comparações e 2922990 movimentações
 Para o vetor de tamanho 8000, temos 61920259 comparações e 12048593 movimentações
 Para o vetor de tamanho 10000, temos 97420257 comparações e 18239451 movimentações
 Para o vetor de tamanho 20000, temos 394140292 comparações e 73388931 movimentações
 A partir do tamanho 50000 estoura o número de comparações.
 Para o vetor de tamanho 50000, temos -1823066735 comparações e 466253672 movimentações
 Para o vetor de tamanho 80000, temos 2024953704 comparações e 1186721042 movimentações
 Para o vetor de tamanho 100000, temos 1301466493 comparações e 1845532687 movimentações



Fica bem próximo da versão randômica no número de comparações, pouca coisa a menos. Função exponencial com $O(n^2)$.



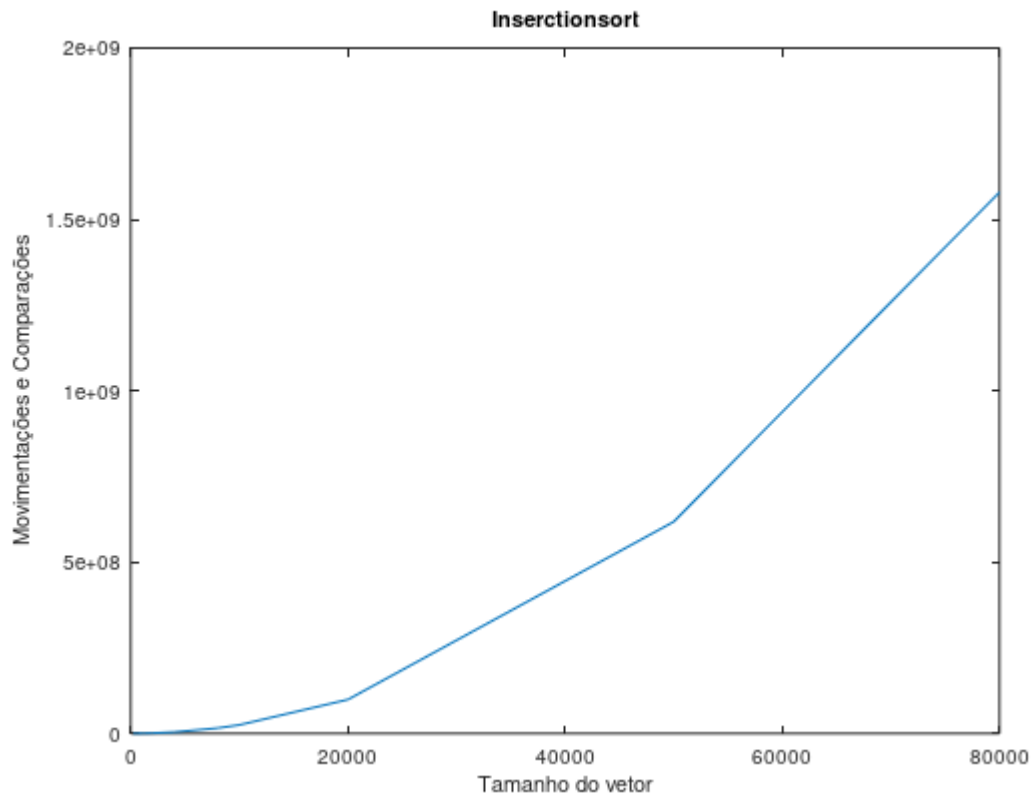
O número de movimentações cai bem comparativamente com o randômico, fica uma mistura do crescente e do randômico. Ficando com metade mais ou menos do randômico. Também é $O(n^2)$.

2) InsertionSort

2.1) Número de comparações e movimentações para um vetor totalmente randômico:

Para o vetor de tamanho 250, temos 16617 comparações e 16617 movimentações
Para o vetor de tamanho 500, temos 61825 comparações e 61825 movimentações
Para o vetor de tamanho 750, temos 141024 comparações e 141024 movimentações
Para o vetor de tamanho 1000, temos 251175 comparações e 251175 movimentações
Para o vetor de tamanho 1250, temos 382757 comparações e 382757 movimentações
Para o vetor de tamanho 1500, temos 557728 comparações e 557728 movimentações
Para o vetor de tamanho 2000, temos 1003868 comparações e 1003868 movimentações
Para o vetor de tamanho 4000, temos 3967769 comparações e 3967769 movimentações
Para o vetor de tamanho 8000, temos 15678368 comparações e 15678368 movimentações
Para o vetor de tamanho 10000, temos 24901534 comparações e 24901534 movimentações
Para o vetor de tamanho 20000, temos 98960122 comparações e 98960122 movimentações
Para o vetor de tamanho 50000, temos 617802106 comparações e 617802106 movimentações
Para o vetor de tamanho 80000, temos 1579350298 comparações e 1579350298 movimentações
A partir do tamanho 100000 estoura o número de comparações e movimentações.
Para o vetor de tamanho 100000, temos -1814284439 comparações e -1814284439 movimentações

Número de comparações e movimentações é o mesmo.

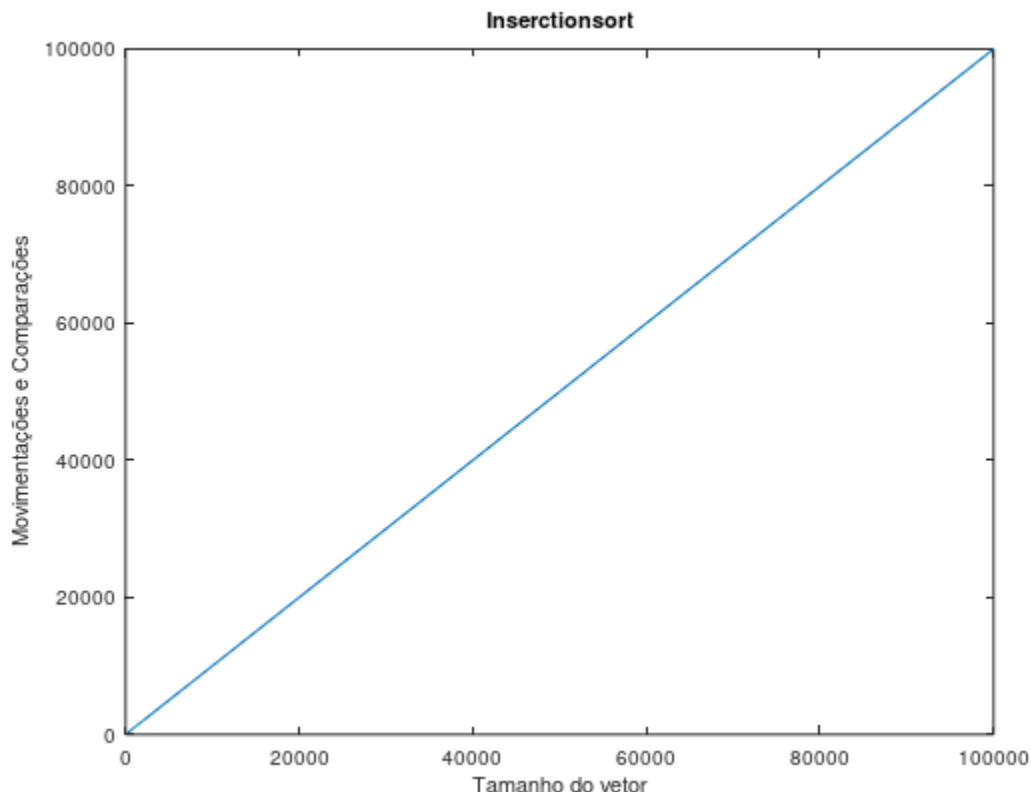


Como o esperado temos uma função exponencial com $O(n^2)$.

2.2) Número de comparações e movimentações para um vetor crescente:

Para o vetor de tamanho 250, temos 249 comparações e 249 movimentações
Para o vetor de tamanho 500, temos 499 comparações e 499 movimentações
Para o vetor de tamanho 750, temos 749 comparações e 749 movimentações
Para o vetor de tamanho 1000, temos 999 comparações e 999 movimentações
Para o vetor de tamanho 1250, temos 1249 comparações e 1249 movimentações
Para o vetor de tamanho 1500, temos 1499 comparações e 1499 movimentações
Para o vetor de tamanho 2000, temos 1999 comparações e 1999 movimentações
Para o vetor de tamanho 4000, temos 3999 comparações e 3999 movimentações
Para o vetor de tamanho 8000, temos 7999 comparações e 7999 movimentações
Para o vetor de tamanho 10000, temos 9999 comparações e 9999 movimentações
Para o vetor de tamanho 20000, temos 19999 comparações e 19999 movimentações
Para o vetor de tamanho 50000, temos 49999 comparações e 49999 movimentações
Para o vetor de tamanho 80000, temos 79999 comparações e 79999 movimentações
Para o vetor de tamanho 100000, temos 99999 comparações e 99999 movimentações

Número de comparações e movimentações é o mesmo.

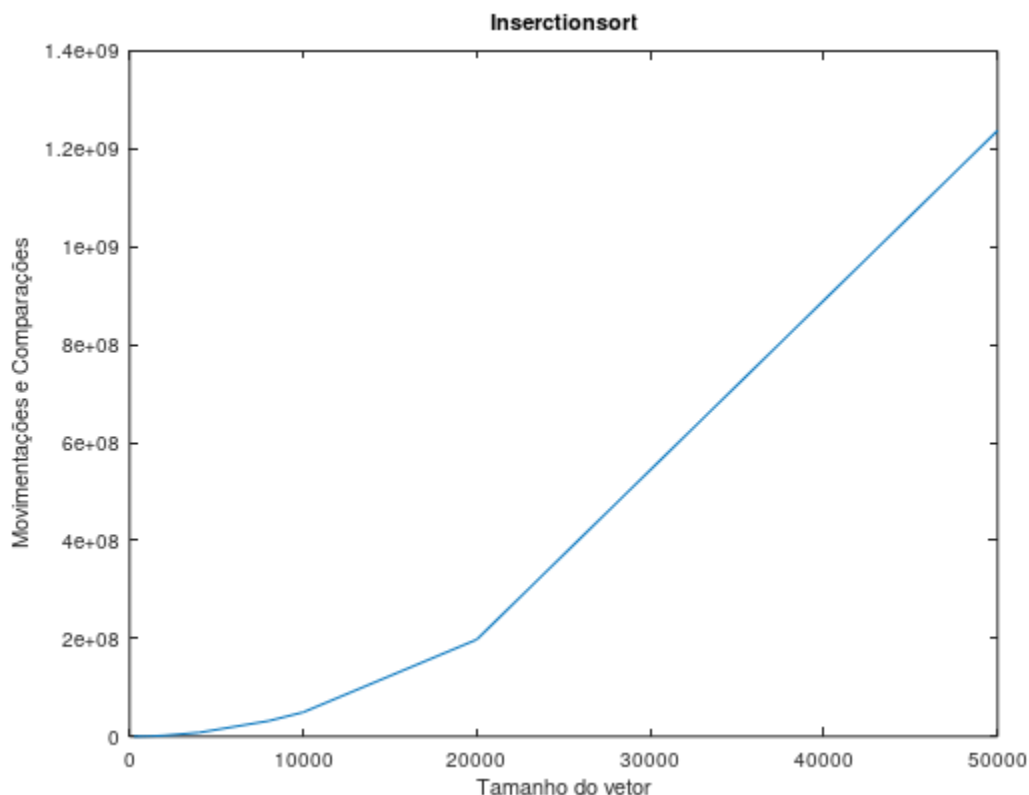


No melhor caso, igual o bubblesort, temos uma função linear com $O(n)$. Porém diferente da bubblesort, a insertionsort também tem movimentação linear e não nula. O que é uma desvantagem comparativamente com a outra.

2.3) Número de comparações e movimentações para um vetor decrescente:

Para o vetor de tamanho 250, temos 31082 comparacoes e 31082 movimentacoes
 Para o vetor de tamanho 500, temos 124000 comparacoes e 124000 movimentacoes
 Para o vetor de tamanho 750, temos 278889 comparacoes e 278889 movimentacoes
 Para o vetor de tamanho 1000, temos 495608 comparacoes e 495608 movimentacoes
 Para o vetor de tamanho 1250, temos 773981 comparacoes e 773981 movimentacoes
 Para o vetor de tamanho 1500, temos 1114724 comparacoes e 1114724 movimentacoes
 Para o vetor de tamanho 2000, temos 1980801 comparacoes e 1980801 movimentacoes
 Para o vetor de tamanho 4000, temos 7922364 comparacoes e 7922364 movimentacoes
 Para o vetor de tamanho 8000, temos 31684437 comparacoes e 31684437 movimentacoes
 Para o vetor de tamanho 10000, temos 49504212 comparacoes e 49504212 movimentacoes
 Para o vetor de tamanho 20000, temos 198010576 comparacoes e 198010576 movimentacoes
 Para o vetor de tamanho 50000, temos 1237525224 comparacoes e 1237525224 movimentacoes
 A partir do tamanho 80000 estoura o número de comparações e movimentações.
 Para o vetor de tamanho 80000, temos -1126929600 comparacoes e -1126929600 movimentacoes
 Para o vetor de tamanho 100000, temos 655088012 comparacoes e 655088012 movimentacoes

Número de comparações e movimentações é o mesmo.



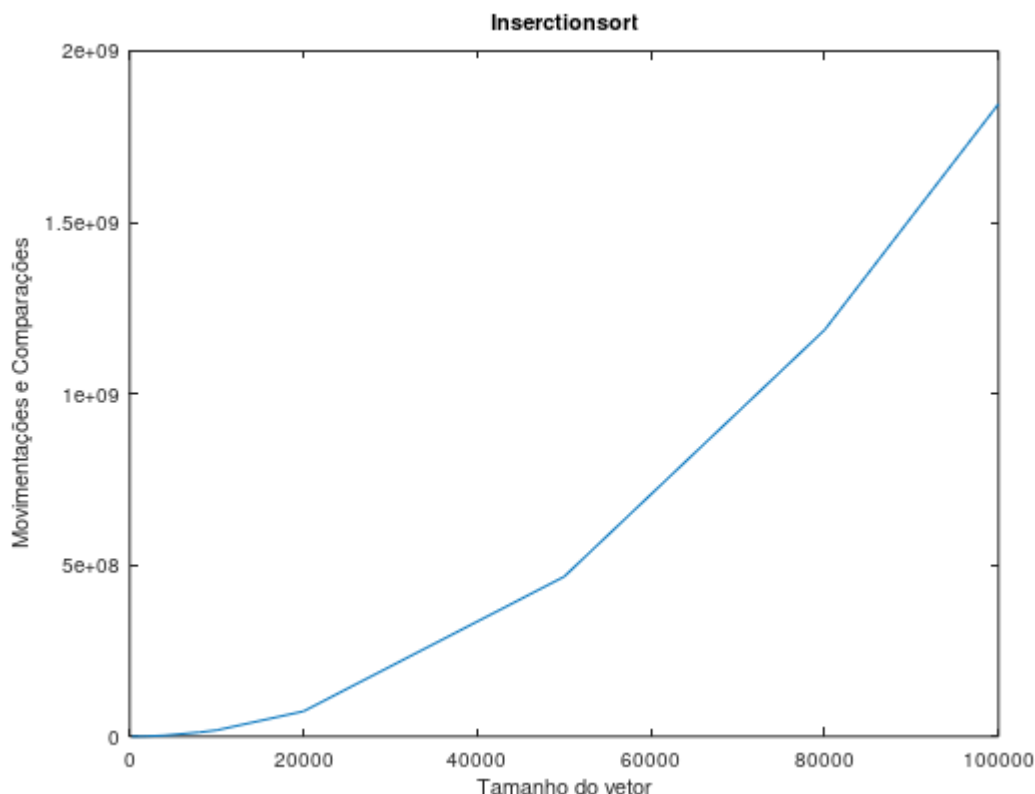
No pior caso ele age como esperado. Função exponencial com $O(n^2)$. Comparativamente com o bubblesort ele estoura um pouco depois nas comparações, mas eles tem a mesma quantidade de movimentações (máxima).

2.4) Número de comparações e movimentações para um vetor metade ordenado crescentemente e metade randômico:

Para o vetor de tamanho 250, temos 12014 comparacoes e 12014 movimentacoes

Para o vetor de tamanho 500, temos 44022 comparações e 44022 movimentações
 Para o vetor de tamanho 750, temos 109323 comparações e 109323 movimentações
 Para o vetor de tamanho 1000, temos 190003 comparações e 190003 movimentações
 Para o vetor de tamanho 1250, temos 290583 comparações e 290583 movimentações
 Para o vetor de tamanho 1500, temos 407003 comparações e 407003 movimentações
 Para o vetor de tamanho 2000, temos 739660 comparações e 739660 movimentações
 Para o vetor de tamanho 4000, temos 2926989 comparações e 2926989 movimentações
 Para o vetor de tamanho 8000, temos 12056592 comparações e 12056592 movimentações
 Para o vetor de tamanho 10000, temos 18249450 comparações e 18249450 movimentações
 Para o vetor de tamanho 20000, temos 73408930 comparações e 73408930 movimentações
 Para o vetor de tamanho 50000, temos 466303671 comparações e 466303671 movimentações
 Para o vetor de tamanho 80000, temos 1186801041 comparações e 1186801041 movimentações
 Para o vetor de tamanho 100000, temos 1845632686 comparações e 1845632686 movimentações

Número de comparações e movimentações é o mesmo.



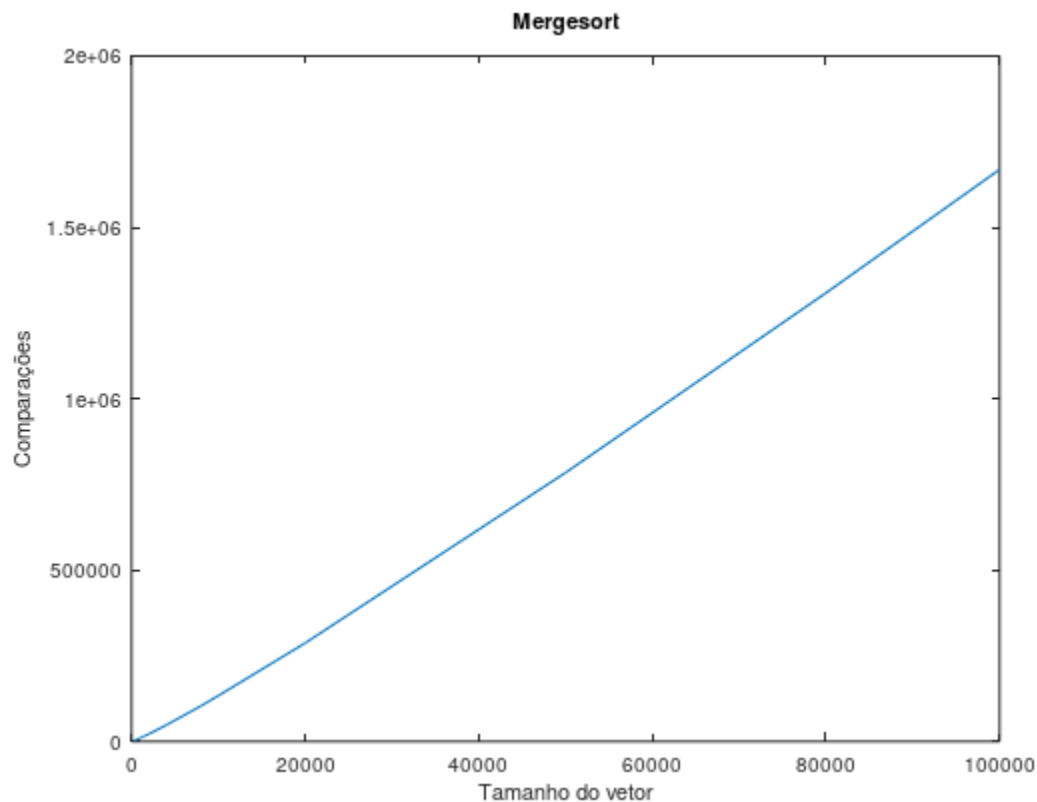
Temos uma mistura do randômico com o crescente. Se comportando bem parecido com o bubblesort. Como uma mistura dos dois.

3) Mergesort

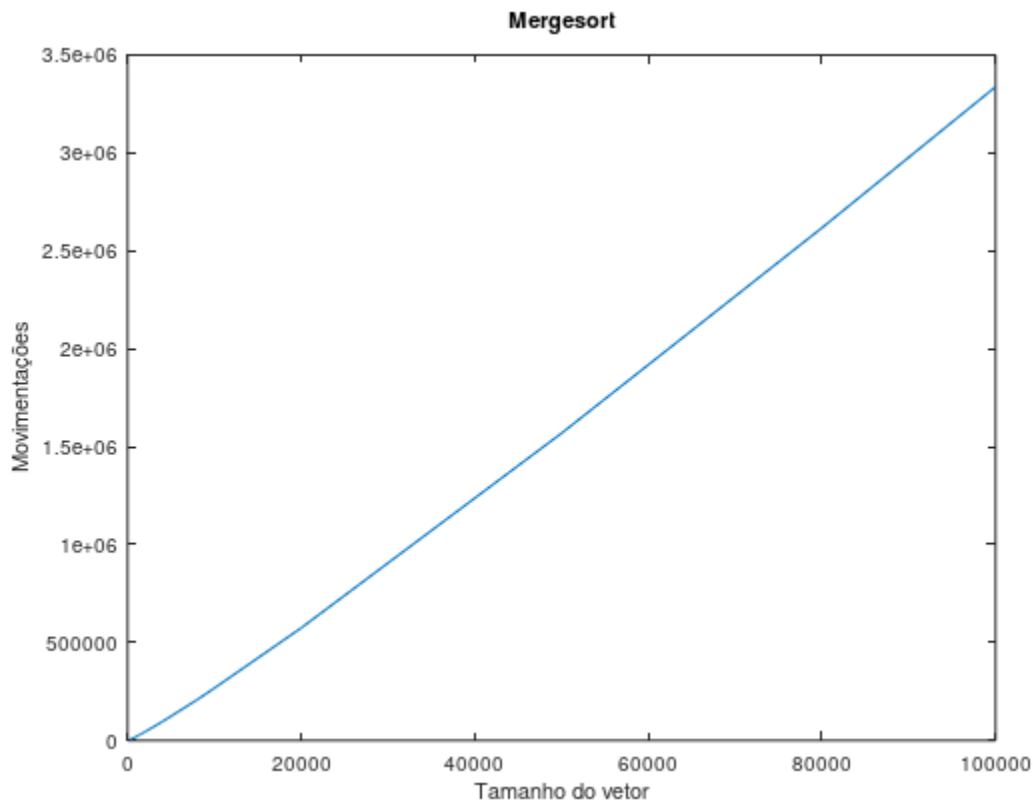
Todos os 4 tipos de teste deram iguais: randômico, crescente, decrescente, metade crescente e metade randômico.

Para o vetor de tamanho 250, temos 1994 comparações e 3988 movimentações
 Para o vetor de tamanho 500, temos 4488 comparações e 8976 movimentações

Para o vetor de tamanho 750, temos 7226 comparações e 14452 movimentações
Para o vetor de tamanho 1000, temos 9976 comparações e 19952 movimentações
Para o vetor de tamanho 1250, temos 12952 comparações e 25904 movimentações
Para o vetor de tamanho 1500, temos 15952 comparações e 31904 movimentações
Para o vetor de tamanho 2000, temos 21952 comparações e 43904 movimentações
Para o vetor de tamanho 4000, temos 47904 comparações e 95808 movimentações
Para o vetor de tamanho 8000, temos 103808 comparações e 207616 movimentações
Para o vetor de tamanho 10000, temos 133616 comparações e 267232 movimentações
Para o vetor de tamanho 20000, temos 287232 comparações e 574464 movimentações
Para o vetor de tamanho 50000, temos 784464 comparações e 1568928 movimentações
Para o vetor de tamanho 80000, temos 1308928 comparações e 2617856 movimentações
Para o vetor de tamanho 100000, temos 1668928 comparações e 3337856 movimentações



O número de comparações é logarítmico, assim o gráfico era o esperado já que $O(n \log n)$ com a junção das duas funções. Sendo a comparação responsável pelo $\log n$.

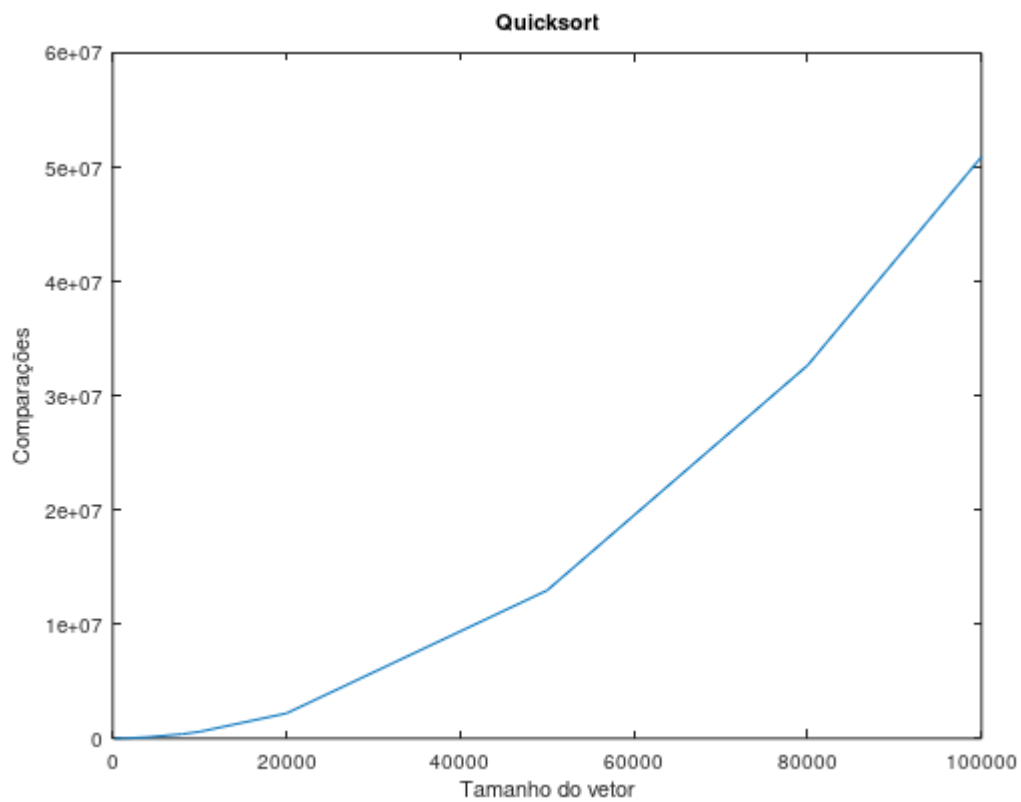


O número de movimentações é linear, dentro do esperado, pois $O(n \log n)$ com a junção das duas funções. Sendo a movimentação responsável pelo n .

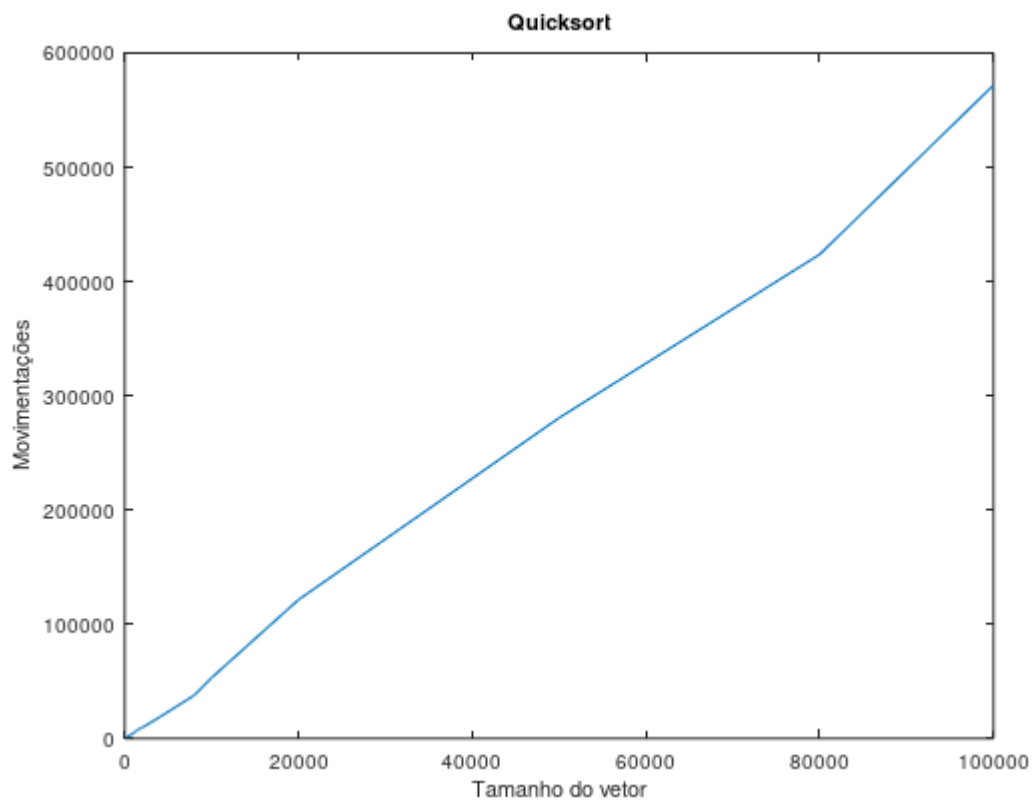
4) Quicksort

4.1) Número de comparações e movimentações para um vetor totalmente randômico:

Para o vetor de tamanho 250, temos 2230 comparações e 1049 movimentações
 Para o vetor de tamanho 500, temos 5900 comparações e 2755 movimentações
 Para o vetor de tamanho 750, temos 8131 comparações e 3256 movimentações
 Para o vetor de tamanho 1000, temos 12894 comparações e 4653 movimentações
 Para o vetor de tamanho 1250, temos 17094 comparações e 5871 movimentações
 Para o vetor de tamanho 1500, temos 24129 comparações e 7711 movimentações
 Para o vetor de tamanho 2000, temos 36038 comparações e 9431 movimentações
 Para o vetor de tamanho 4000, temos 112793 comparações e 18583 movimentações
 Para o vetor de tamanho 8000, temos 385066 comparações e 37789 movimentações
 Para o vetor de tamanho 10000, temos 588562 comparações e 53109 movimentações
 Para o vetor de tamanho 20000, temos 2199824 comparações e 121430 movimentações
 Para o vetor de tamanho 50000, temos 12963728 comparações e 280685 movimentações
 Para o vetor de tamanho 80000, temos 32673026 comparações e 423783 movimentações
 Para o vetor de tamanho 100000, temos 50919157 comparações e 571871 movimentações



O pior caso do quicksort é $O(n^2)$, quando o pivô está num dos extremos. Como o gráfico ficou exponencial, deve ser um caso desse.



As movimentações ficaram logarítmicas. Então o gráfico quase $n \log n$, se encaixa no padrão.

4.2) Número de comparações e movimentações para um vetor crescente:

Para o vetor de tamanho 250, temos 27400 comparações e 250 movimentações

Acima disso estourou o ponteiro:

`munmap_chunk(): invalid pointer`

Aborted (core dumped)

Partes ordenadas crescentemente dão erro com esse algoritmo

4.3) Número de comparações e movimentações para um vetor decrescente:

Para o vetor de tamanho 250, temos 17271 comparações e 6544 movimentações

Para o vetor de tamanho 500, temos 30517 comparações e 14050 movimentações

Para o vetor de tamanho 750, temos 66724 comparações e 32826 movimentações

Para o vetor de tamanho 1000, temos 50191 comparações e 22732 movimentações

Para o vetor de tamanho 1250, temos 120221 comparações e 57696 movimentações

Para o vetor de tamanho 1500, temos 66928 comparações e 29197 movimentações

Para o vetor de tamanho 2000, temos 92036 comparações e 36260 movimentações

Para o vetor de tamanho 4000, temos 319943 comparações e 99737 movimentações

Para o vetor de tamanho 8000, temos 999791 comparações e 330853 movimentações

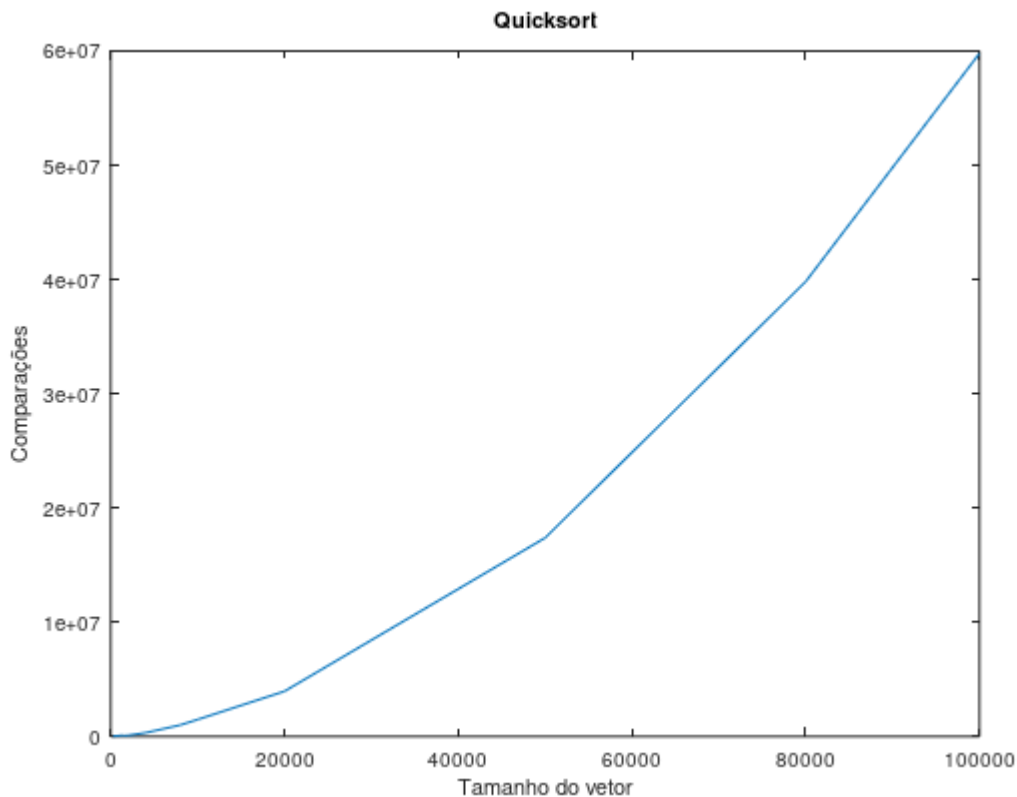
Para o vetor de tamanho 10000, temos 1482036 comparações e 504273 movimentações

Para o vetor de tamanho 20000, temos 3956080 comparações e 1006520 movimentações

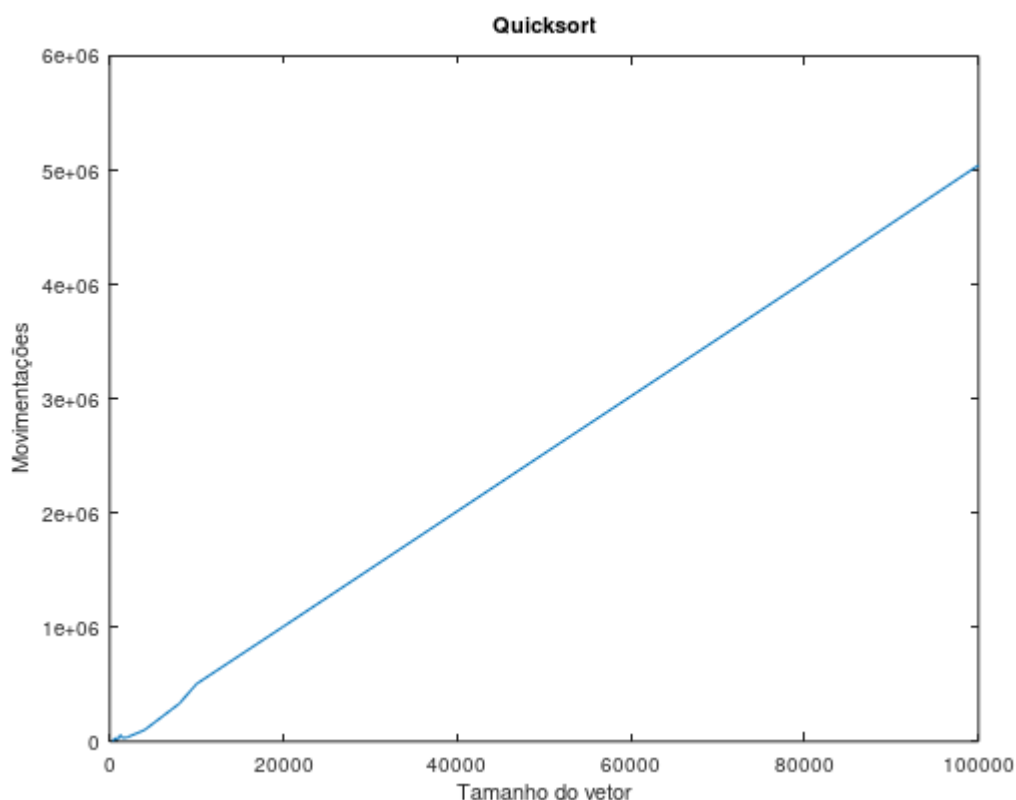
Para o vetor de tamanho 50000, temos 17387668 comparações e 2519914 movimentações

Para o vetor de tamanho 80000, temos 39822390 comparações e 4027784 movimentações

Para o vetor de tamanho 100000, temos 59798230 comparações e 5044523 movimentações



A função ficou exponencial, dentro do esperado, pois como é decrescente, o pivô fica na extremidade, então cai sempre no pior caso.



As movimentações ficaram logarítmicas. Então o gráfico quase $n \log n$, se encaixa no padrão.

4.4) Número de comparações e movimentações para um vetor metade ordenado crescentemente e metade randômico:

Para o vetor de tamanho 250, temos 2945 comparações e 771 movimentações

Para o vetor de tamanho 500, temos 14974 comparações e 1444 movimentações

Para o vetor de tamanho 750, temos 15368 comparações e 2743 movimentações

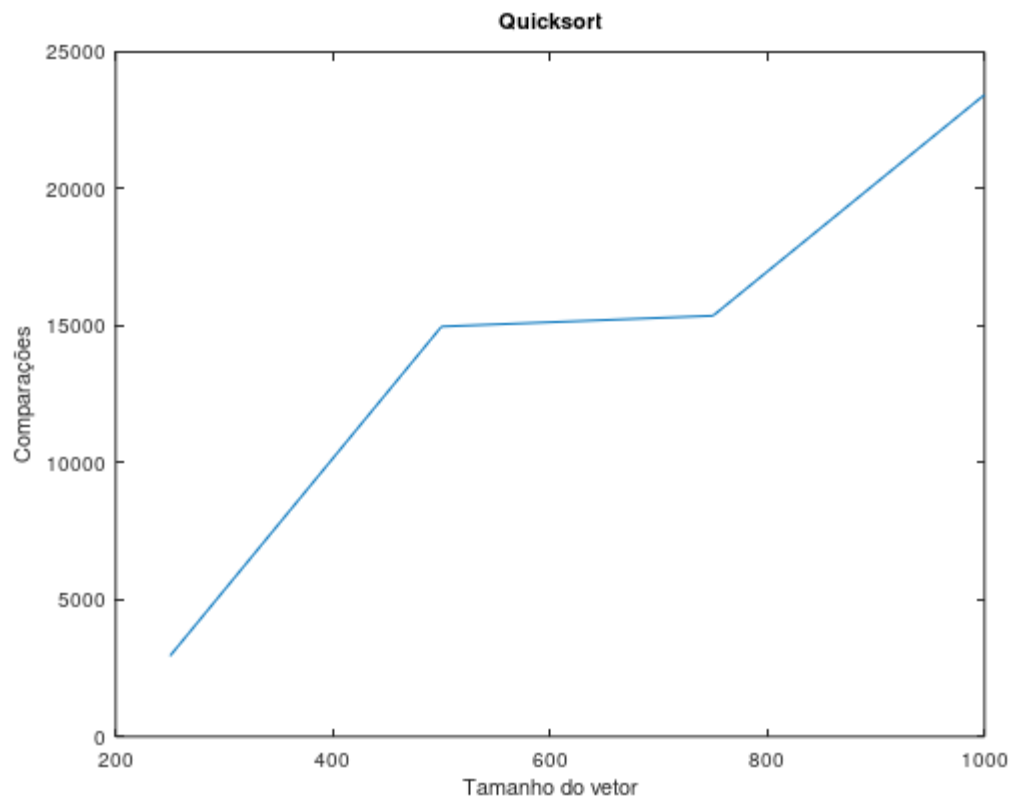
Para o vetor de tamanho 1000, temos 23439 comparações e 3795 movimentações

Acima disso estourou o ponteiro:

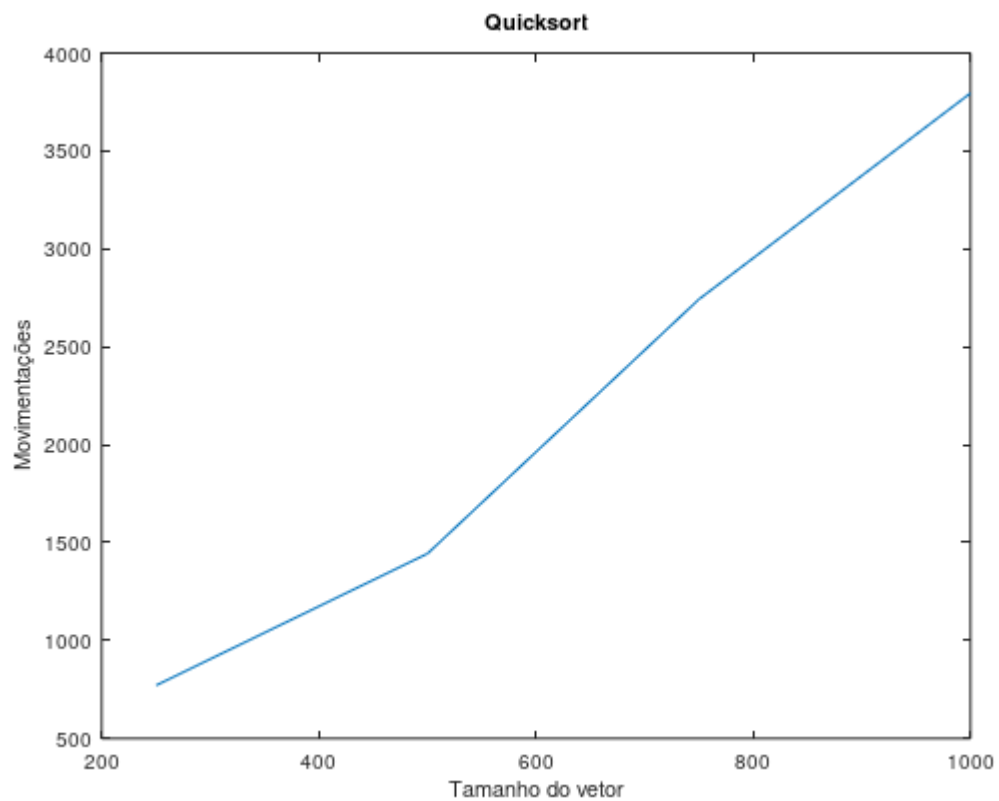
`munmap_chunk(): invalid pointer`

Aborted (core dumped)

Partes ordenadas crescentemente dão erro com esse algoritmo



Não saiu nada do planejado. O gráfico deveria ser logarítmico.

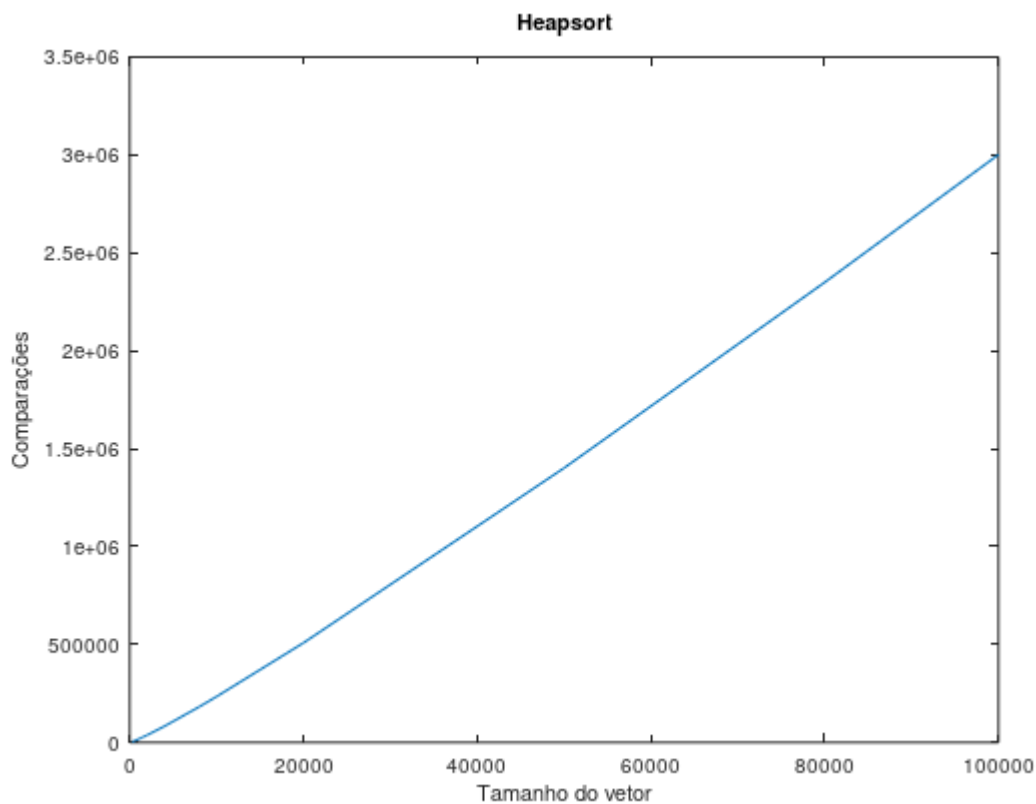


As movimentações ficaram logarítmicas. Então o gráfico quase $n \log n$, se encaixa no padrão.

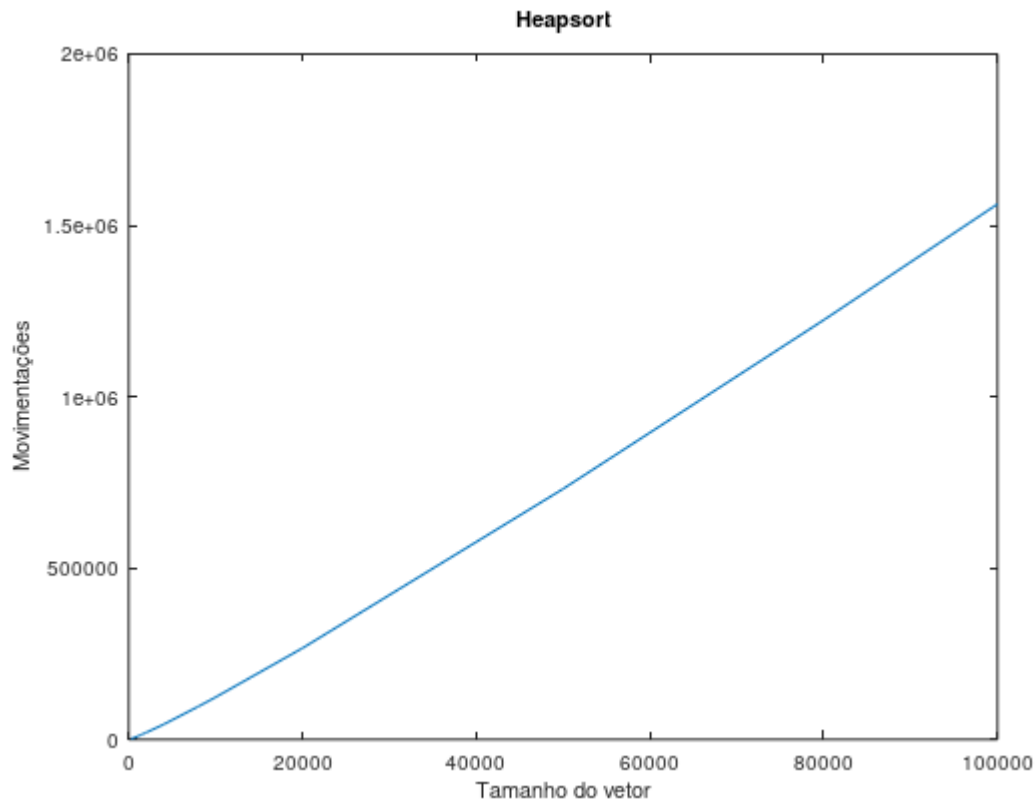
5) Heapsort

5.1) Número de comparações e movimentações para um vetor totalmente randômico:

Para o vetor de tamanho 250, temos 3210 comparações e 1755 movimentações
Para o vetor de tamanho 500, temos 7394 comparações e 4005 movimentações
Para o vetor de tamanho 750, temos 12012 comparações e 6446 movimentações
Para o vetor de tamanho 1000, temos 16788 comparações e 8995 movimentações
Para o vetor de tamanho 1250, temos 21946 comparações e 11767 movimentações
Para o vetor de tamanho 1500, temos 27088 comparações e 14469 movimentações
Para o vetor de tamanho 2000, temos 37586 comparações e 20002 movimentações
Para o vetor de tamanho 4000, temos 82892 comparações e 43840 movimentações
Para o vetor de tamanho 8000, temos 182128 comparações e 96058 movimentações
Para o vetor de tamanho 10000, temos 234358 comparações e 123362 movimentações
Para o vetor de tamanho 20000, temos 508538 comparações e 266508 movimentações
Para o vetor de tamanho 50000, temos 1401676 comparações e 731305 movimentações
Para o vetor de tamanho 80000, temos 2349856 comparações e 1224044 movimentações



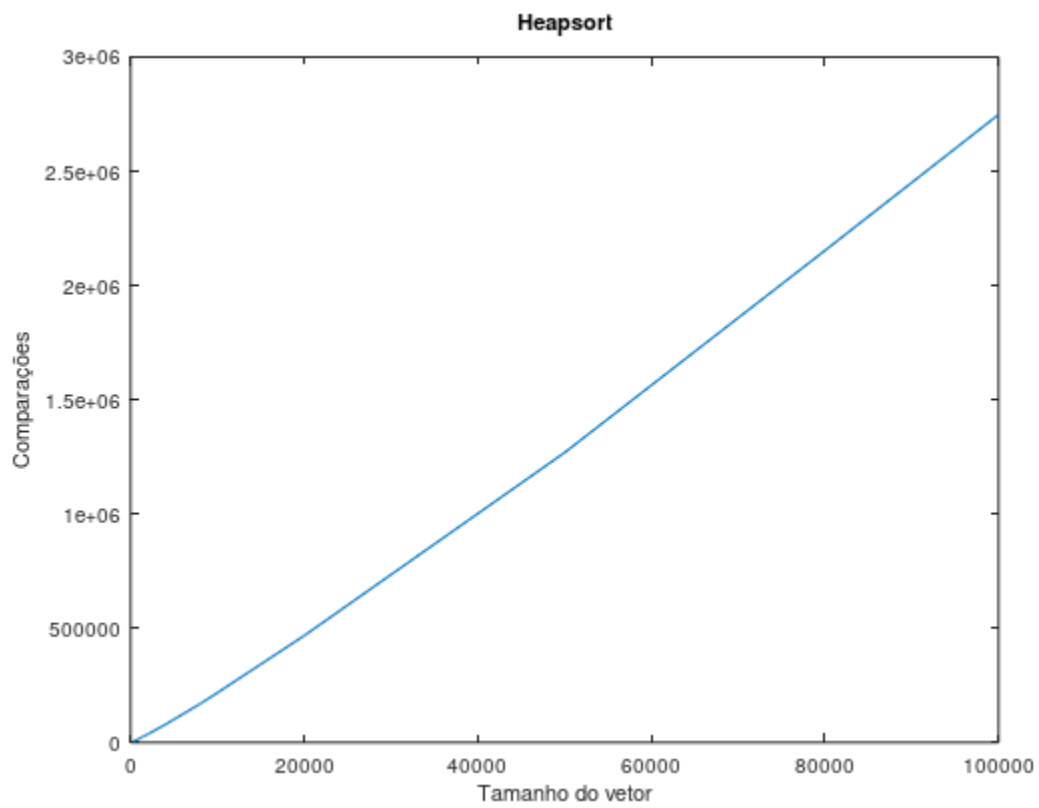
Se comporta como o esperado, Heapsort é $O(n \log n)$ sempre.



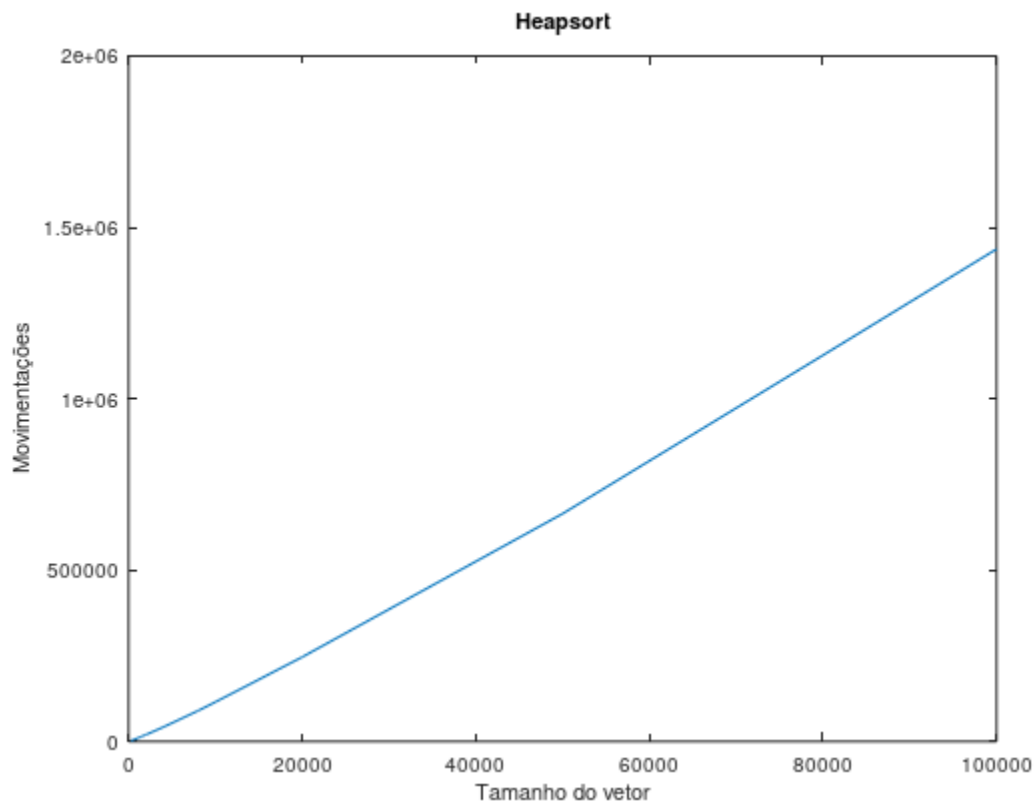
Se comporta como o esperado, Heapsort é $O(n \log n)$ sempre.

5.2) Número de comparações e movimentações para um vetor crescente:

Para o vetor de tamanho 250, temos 3368 comparações e 1893 movimentações
Para o vetor de tamanho 500, temos 7718 comparações e 4286 movimentações
Para o vetor de tamanho 750, temos 12348 comparações e 6788 movimentações
Para o vetor de tamanho 1000, temos 17058 comparações e 9369 movimentações
Para o vetor de tamanho 1250, temos 21882 comparações e 11916 movimentações
Para o vetor de tamanho 1500, temos 26994 comparações e 14674 movimentações
Para o vetor de tamanho 2000, temos 36492 comparações e 19740 movimentações
Para o vetor de tamanho 4000, temos 78430 comparações e 41977 movimentações
Para o vetor de tamanho 8000, temos 168792 comparações e 89604 movimentações
Para o vetor de tamanho 10000, temos 217334 comparações e 115109 movimentações
Para o vetor de tamanho 20000, temos 467288 comparações e 246329 movimentações
Para o vetor de tamanho 50000, temos 1268564 comparações e 664631 movimentações
Para o vetor de tamanho 80000, temos 2153254 comparações e 1127062 movimentações
Para o vetor de tamanho 100000, temos 2748166 comparações e 1436859 movimentações



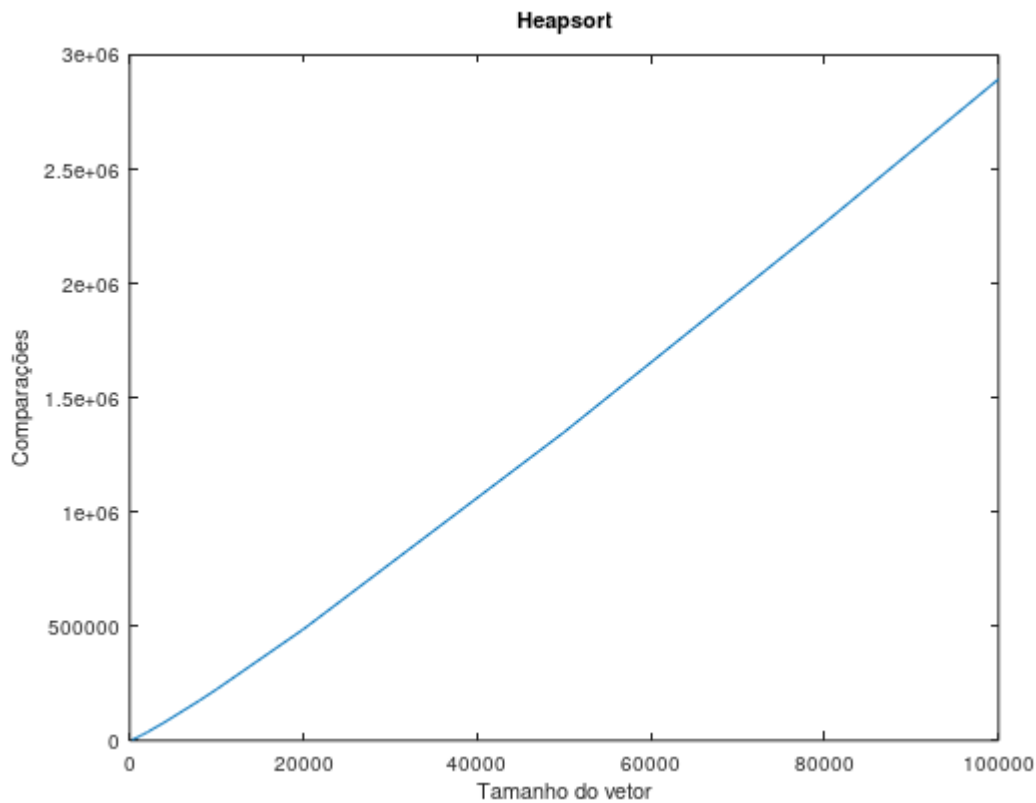
Se comporta como o esperado, Heapsort é $O(n \log n)$ sempre.



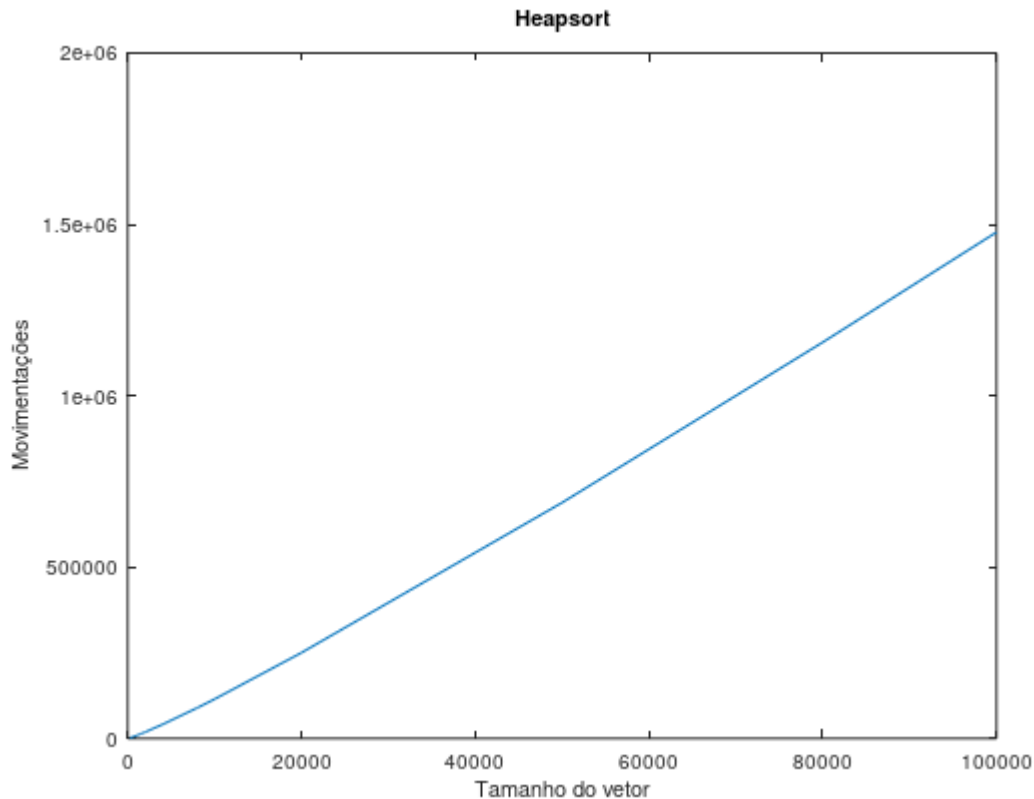
Se comporta como o esperado, Heapsort é $O(n \log n)$ sempre.

5.3) Número de comparações e movimentações para um vetor decrescente:

Para o vetor de tamanho 250, temos 3020 comparações e 1594 movimentações
Para o vetor de tamanho 500, temos 6986 comparações e 3637 movimentações
Para o vetor de tamanho 750, temos 11404 comparações e 5915 movimentações
Para o vetor de tamanho 1000, temos 15898 comparações e 8246 movimentações
Para o vetor de tamanho 1250, temos 20766 comparações e 10787 movimentações
Para o vetor de tamanho 1500, temos 25736 comparações e 13424 movimentações
Para o vetor de tamanho 2000, temos 35822 comparações e 18568 movimentações
Para o vetor de tamanho 4000, temos 79524 comparações e 40930 movimentações
Para o vetor de tamanho 8000, temos 174076 comparações e 89341 movimentações
Para o vetor de tamanho 10000, temos 223986 comparações e 114839 movimentações
Para o vetor de tamanho 20000, temos 487250 comparações e 250283 movimentações
Para o vetor de tamanho 50000, temos 1349452 comparações e 688216 movimentações
Para o vetor de tamanho 80000, temos 2265630 comparações e 1156165 movimentações
Para o vetor de tamanho 100000, temos 2894466 comparações e 1477340 movimentações



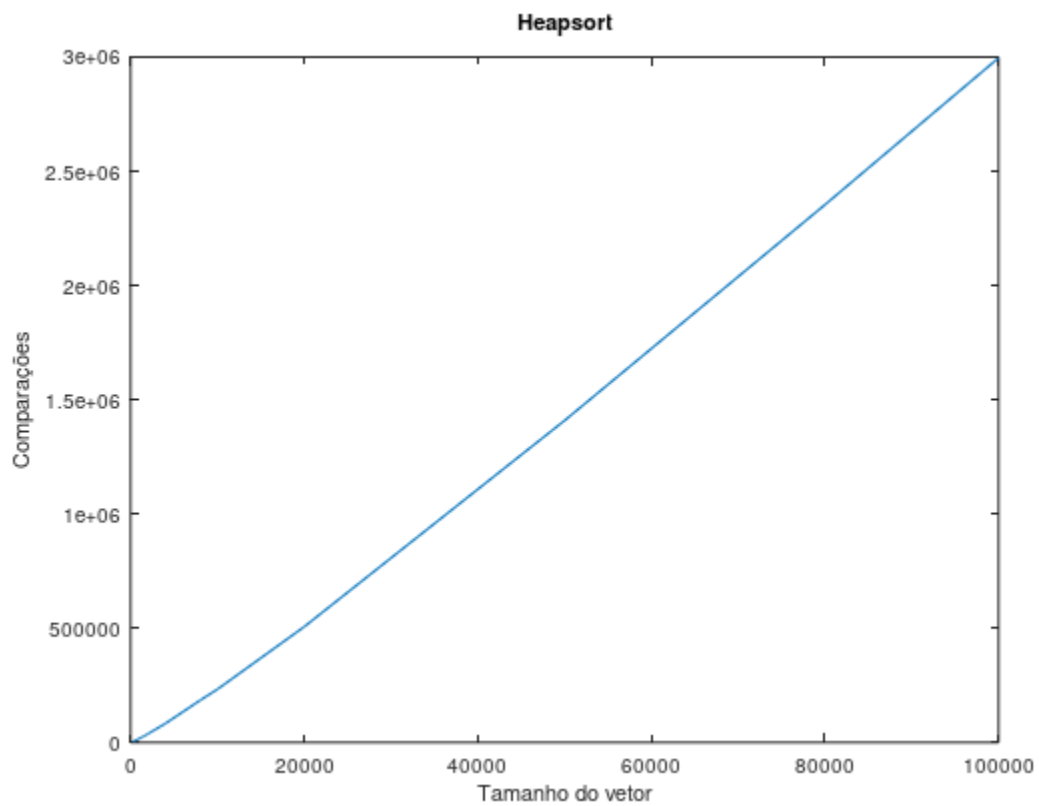
Se comporta como o esperado, Heapsort é $O(n \log n)$ sempre.



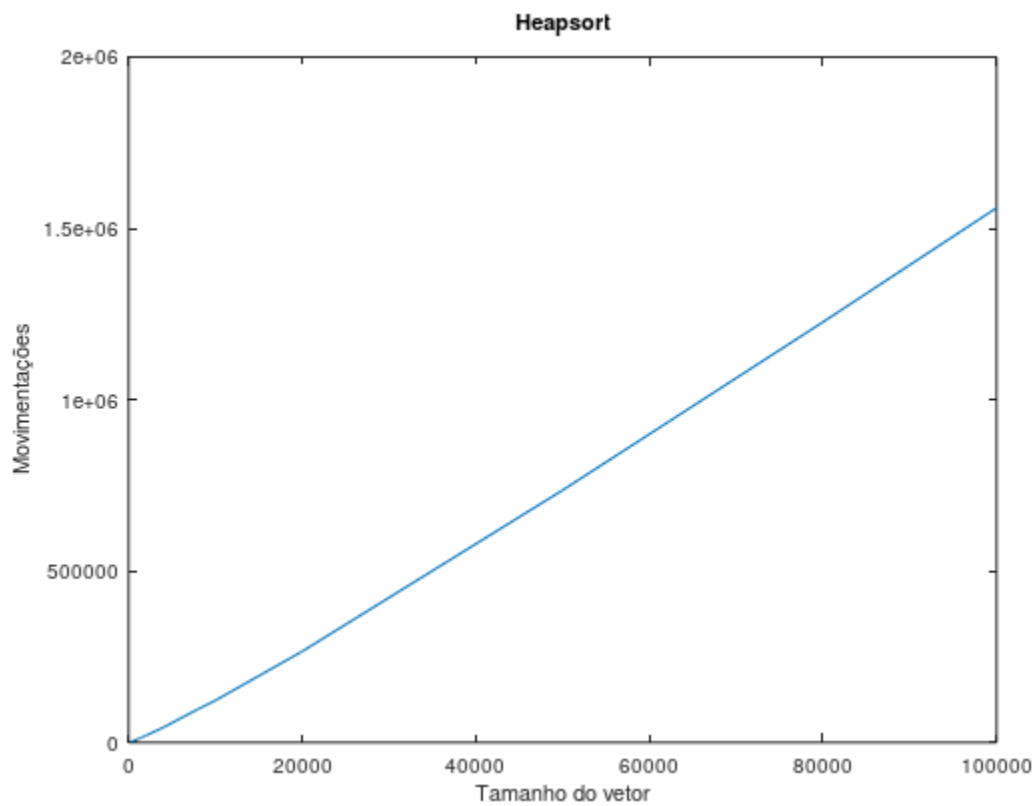
Se comporta como o esperado, Heapsort é $O(n \log n)$ sempre.

4.4) Número de comparações e movimentações para um vetor metade ordenado crescentemente e metade randômico:

Para o vetor de tamanho 250, temos 3296 comparações e 1822 movimentações
Para o vetor de tamanho 500, temos 7364 comparações e 4021 movimentações
Para o vetor de tamanho 750, temos 12358 comparações e 6674 movimentações
Para o vetor de tamanho 1000, temos 17140 comparações e 9240 movimentações
Para o vetor de tamanho 1250, temos 21934 comparações e 11802 movimentações
Para o vetor de tamanho 1500, temos 26918 comparações e 14399 movimentações
Para o vetor de tamanho 2000, temos 37738 comparações e 20137 movimentações
Para o vetor de tamanho 4000, temos 82596 comparações e 43811 movimentações
Para o vetor de tamanho 8000, temos 184318 comparações e 97349 movimentações
Para o vetor de tamanho 10000, temos 232690 comparações e 122597 movimentações
Para o vetor de tamanho 20000, temos 506340 comparações e 265706 movimentações
Para o vetor de tamanho 50000, temos 1409044 comparações e 736041 movimentações
Para o vetor de tamanho 80000, temos 2353740 comparações e 1226803 movimentações
Para o vetor de tamanho 100000, temos 2996182 comparações e 1559894 movimentações



Se comporta como o esperado, Heapsort é $O(n \log n)$ sempre.



Se comporta como o esperado, Heapsort é $O(n \log n)$ sempre.

Os quatro testes no Heapsort se comportam bem semelhantemente. Ele é bem consistente, não importando o tipo de vetor.

Conclusões

Analisando os 5 algoritmos podemos notar alguns pontos onde cada um se sobressai. O Bubblesort e o Insertionsort são os menos eficazes, pois estouram mais rápido e tem grande variação conforme o tipo de vetor usado. Entre os dois o Insertionsort consegue ter uma pequena vantagem. Mas os dois são os únicos que fazem o mínimo de movimentos e comparações quando o vetor já está ordenado, apenas n .

O Mergesort é absurdamente exato. Ele sempre dá a mesma quantidade não importa o vetor utilizado, sua consistência é assustadora. E o número de comparações é mais baixo, na ordem de $n \log n$.

O Quicksort só se comporta bem com casos aleatórios. Em casos crescentes ele quebra e decrescentes ele funciona pior que o Bubble e o Insertion. É o mais inconsistente de todos.

O Heapsort tem boa consistência. Não é tão exato como o Merge, mas fica bem próximo dos valores de comparações em todos os tipos de vetores. Fica sempre em $n \log n$.