
Laboratório 1 - Algoritmos de ordenação

Relatório

Instituto de computação - UNICAMP

Campinas, 2 de outubro de 2013

Raíssa Costa Machado 097081

Matheus Ferreira Tavares Boy 103501

Sumário

1	Introdução	2
2	Modelagem	2
3	Estrutura	2
4	Resultados	2
5	Conclusões	2

Best #	Tempo (microssegundos)	Worst #	Tempo (microssegundos)
0	1.2	0	94.59
1	1.24	1	94.71
2	1.21	2	94.6
3	1.21	3	94.5
4	1.22	4	94.52
5	1.21	5	94.49
6	1.22	6	94.55
7	1.21	7	94.53
8	1.21	8	94.53
9	1.22	9	94.49
Média	1.215	Média	94.551

Tabela 1: Tempos de execução dos piores e melhores casos do Bubble Sort

1 Introdução

A proposta deste laboratório é a análise prática de algoritmos de ordenação bem conhecidos: Insertion, Bubble, Merge, Quick e Heapsort. Implementamos os programas em C++ para cada um desses algoritmos e testamos e avaliamos seus desempenhos, do ponto de vista dos tempos de execução, através de um conjunto de dados contendo 50 vetores aleatórios com uma variação de 20 a 1000 elementos com intervalos de 20. Para a análise dos piores e melhores casos usamos 10 vetores com 100 elementos.

2 Modelagem

3 Estrutura

4 Resultados

5 Conclusões

Referências

- [1] Rodolfo Azevedo, Bruno Albertini, and Sandro Rigo. Arp: Um gerenciador de pacotes para sistemas embarcados com processadores modelados

Best #	Tempo (microsegundos)	Worst #	Tempo (microsegundos)
0	15.39	0	17.76
1	15.79	1	17.52
2	15.78	2	17.46
3	15.9	3	17.7
4	15.31	4	16.99
5	15.69	5	17.49
6	15.45	6	17.26
7	15.56	7	17.5
8	15.6	8	17.25
9	15.59	9	17.42
Média	15.606	Média	17.435

Tabela 2: Tempos de execução dos piores e melhores casos do Heapsort

Best #	Tempo (microsegundos)	Worst #	Tempo (microsegundos)
0	1.77	0	42.46
1	1.77	1	42.47
2	1.78	2	42.48
3	1.78	3	42.47
4	1.78	4	42.47
5	1.78	5	42.46
6	1.78	6	42.47
7	1.78	7	42.47
8	1.76	8	42.47
9	1.76	9	42.47
Média	1.774	Média	42.469

Tabela 3: Tempos de execução dos piores e melhores casos do Insertion Sort

Best #	Tempo (microsegundos)	Worst #	Tempo (microsegundos)
0	13.98	0	14.78
1	14.01	1	14.77
2	14.02	2	14.78
3	14.03	3	14.78
4	14.02	4	14.78
5	13.97	5	14.78
6	13.98	6	14.76
7	13.96	7	14.72
8	14.19	8	14.78
9	14.15	9	14.76
Média	14.031	Média	14.769

Tabela 4: Tempos de execução dos piores e melhores casos do Merge Sort

Best #	Tempo (microssegundos)	Worst #	Tempo (microssegundos)
0	10.09	0	41.35
1	11.51	1	41.41
2	10.55	2	41.46
3	10.98	3	41.4
4	10.19	4	41.38
5	10.13	5	41.36
6	10.63	6	41.54
7	11.35	7	41.36
8	11.07	8	41.39
9	10.98	9	41.44
Média	10.748	Média	41.409

Tabela 5: Tempos de execução dos piores e melhores casos do Quicksort

- em archc. In *Workshop de Sistemas Embarcados - WSE*. SBC, 2010. In Portuguese.
- [2] Rodolfo Azevedo, Sandro Rigo, Marcus Bartholomeu, Guido Araujo, Cristiano Araujo, and Edna Barros. The archc architecture description language and tools. *International Journal of Parallel Programming*, 33:453–484, 2005. 10.1007/s10766-005-7301-0.
 - [3] Christian Bienia, Sanjeev Kumar, Jaswinder Pal Singh, and Kai Li. The parsec benchmark suite: characterization and architectural implications. In *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, pages 72–81. ACM, 2008.
 - [4] Jason Clemons, Haishan Zhu, Silvio Savarese, and Todd Austin. Mevbench: A mobile computer vision benchmarking suite. In *Workload Characterization (IISWC), 2011 IEEE International Symposium on*, pages 91–102. IEEE, 2011.
 - [5] Matthew R Guthaus, Jeffrey S Ringenberg, Dan Ernst, Todd M Austin, Trevor Mudge, and Richard B Brown. Mibench: A free, commercially representative embedded benchmark suite. In *Workload Characterization, 2001. WWC-4. 2001 IEEE International Workshop on*, pages 3–14. IEEE, 2001.
 - [6] Syed Muhammad Zeeshan Iqbal, Yuchen Liang, and Håkan Grahm. Parmibench-an open-source benchmark for embedded multiprocessor systems. *Computer Architecture Letters*, 9(2):45–48, 2010.
 - [7] Chunho Lee, Miodrag Potkonjak, and William H Mangione-Smith. Media-bench: a tool for evaluating and synthesizing multimedia and communications systems. In *Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture*, pages 330–335. IEEE Computer Society, 1997.
 - [8] Stan Liao, Grant Martin, Stuart Swan, and Thorsten Grötker. *System design with SystemC*. Kluwer Academic Pub, 2002.

Random Size	Bubble	Heap	Insertion	Merge	Quick
20	0	0	0	0	0
40	0.01	0	0	0	0
60	0.02	0.01	0.01	0.01	0
80	0.06	0.01	0.02	0.01	0.01
100	0.08	0.02	0.02	0.02	0.01
120	0.11	0.02	0.02	0.02	0.02
140	0.15	0.03	0.04	0.02	0.01
160	0.19	0.03	0.05	0.03	0.02
180	0.26	0.03	0.05	0.03	0.02
200	0.29	0.05	0.07	0.04	0.03
220	0.38	0.04	0.08	0.04	0.03
240	0.43	0.05	0.09	0.04	0.03
260	0.52	0.06	0.1	0.05	0.03
280	0.61	0.06	0.13	0.05	0.04
300	0.73	0.07	0.14	0.06	0.03
320	0.87	0.08	0.17	0.06	0.05
340	0.9	0.07	0.18	0.06	0.04
360	1.01	0.09	0.2	0.07	0.05
380	1.05	0.09	0.22	0.07	0.05
400	1.25	0.09	0.24	0.08	0.06
420	1.38	0.11	0.27	0.08	0.06
440	1.41	0.1	0.28	0.09	0.06
460	1.66	0.12	0.32	0.09	0.07
480	1.75	0.12	0.34	0.1	0.07
500	1.9	0.12	0.38	0.1	0.07
520	2.07	0.13	0.39	0.1	0.08
540	2.24	0.14	0.45	0.11	0.09
560	2.36	0.14	0.44	0.11	0.09
580	2.62	0.15	0.5	0.12	0.08
600	2.79	0.15	0.52	0.13	0.09
620	3.01	0.16	0.55	0.12	0.09
640	3.08	0.17	0.6	0.14	0.1
660	3.27	0.17	0.63	0.13	0.1
680	3.47	0.18	0.66	0.14	0.1
700	3.78	0.18	0.72	0.15	0.11
720	3.99	0.19	0.76	0.15	0.11
740	4.35	0.2	0.78	0.15	0.12
760	4.32	0.2	0.82	0.16	0.11
780	4.67	0.21	0.89	0.16	0.12
800	4.95	0.22	0.93	0.17	0.13
820	4.99	0.22	0.96	0.18	0.13
840	5.45	0.23	1.04	0.17	0.14
860	5.75	0.23	1.08	0.19	0.14
880	6	0.24	1.13	0.19	0.13
900	6.26	0.25	1.18	0.19	0.15
920	6.46	0.25	1.18	0.2	0.14
940	6.87	0.26	1.3	0.2	0.15
960	7.27	0.26	1.3	0.21	0.16
980	7.56	0.27	1.36	0.22	0.15
1000	7.8	0.27	1.44	0.22	0.16

Tabela 6: Tempos de execução dos vetores aleatórios de todos os sorts

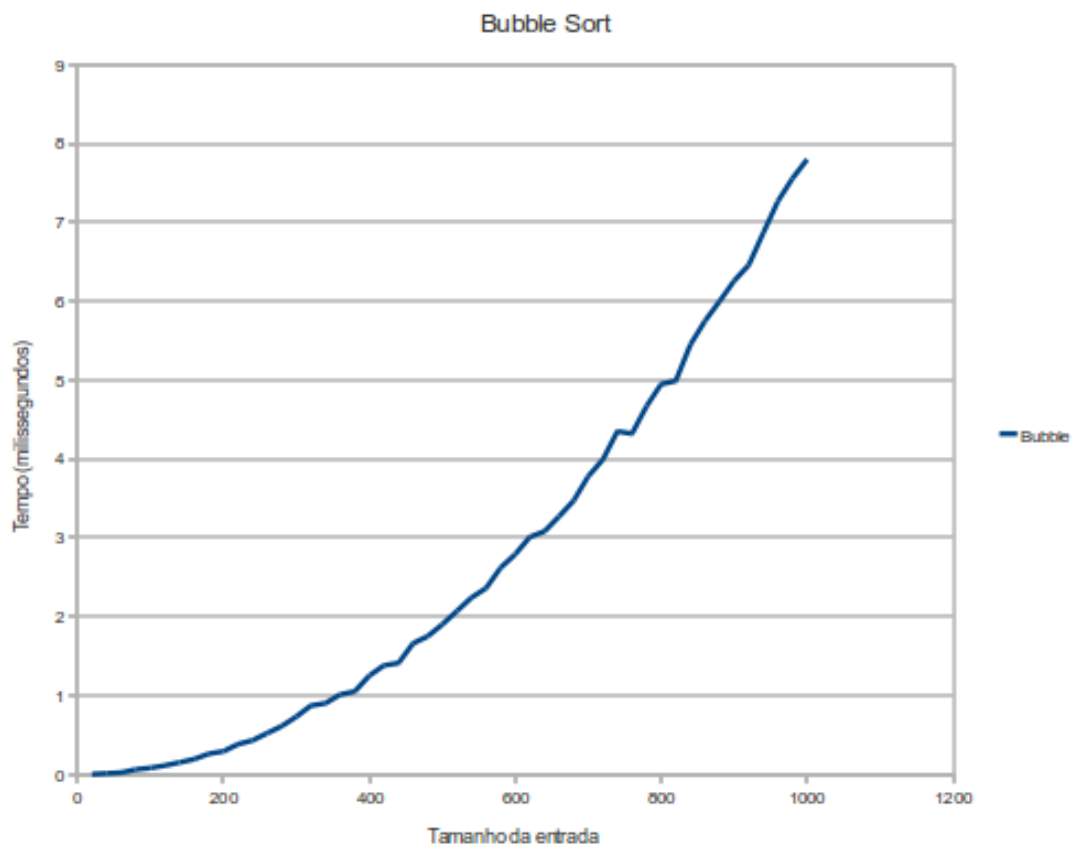


Figura 1: Gráfico de desempenho do Bubble Sort

- [9] S. Rigo, G. Araujo, M. Bartholomeu, and R. Azevedo. Archc: a systemc-based architecture description language. In *16th Symposium on Computer Architecture and High Performance Computing, 2004 - SBAC-PAD 2004*, pages 66 – 73, 2004. Best Paper Award.
- [10] Steven Cameron Woo, Moriyoshi Ohara, Evan Torrie, Jaswinder Pal Singh, and Anoop Gupta. The splash-2 programs: Characterization and methodological considerations. In *ACM SIGARCH Computer Architecture News*, volume 23, pages 24–36. ACM, 1995.

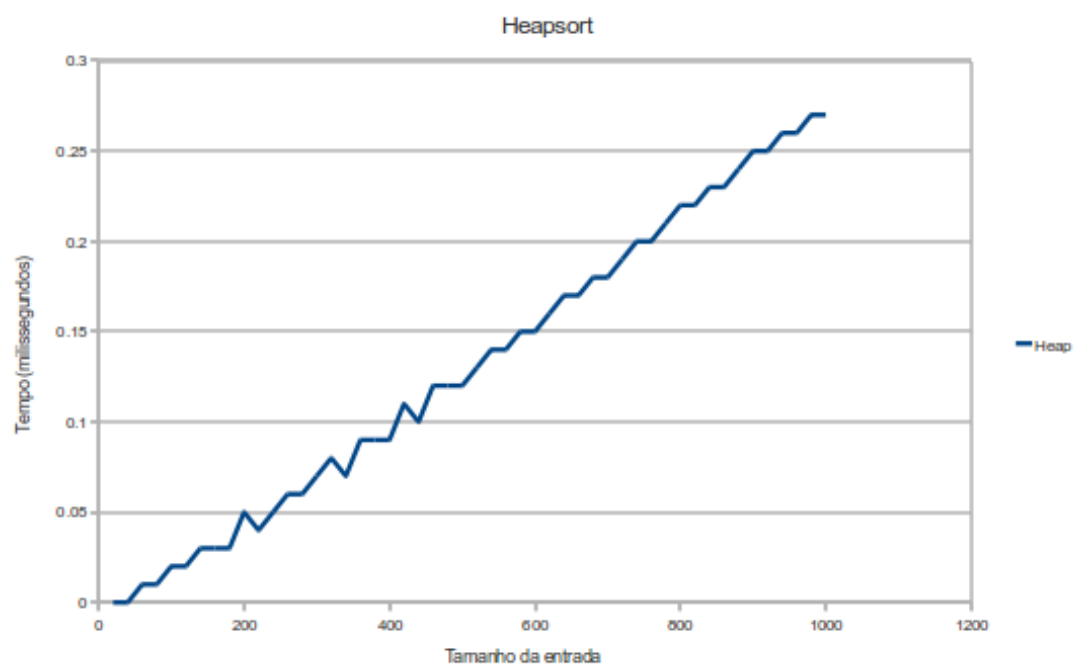


Figura 2: Gráfico de desempenho do Heapsort

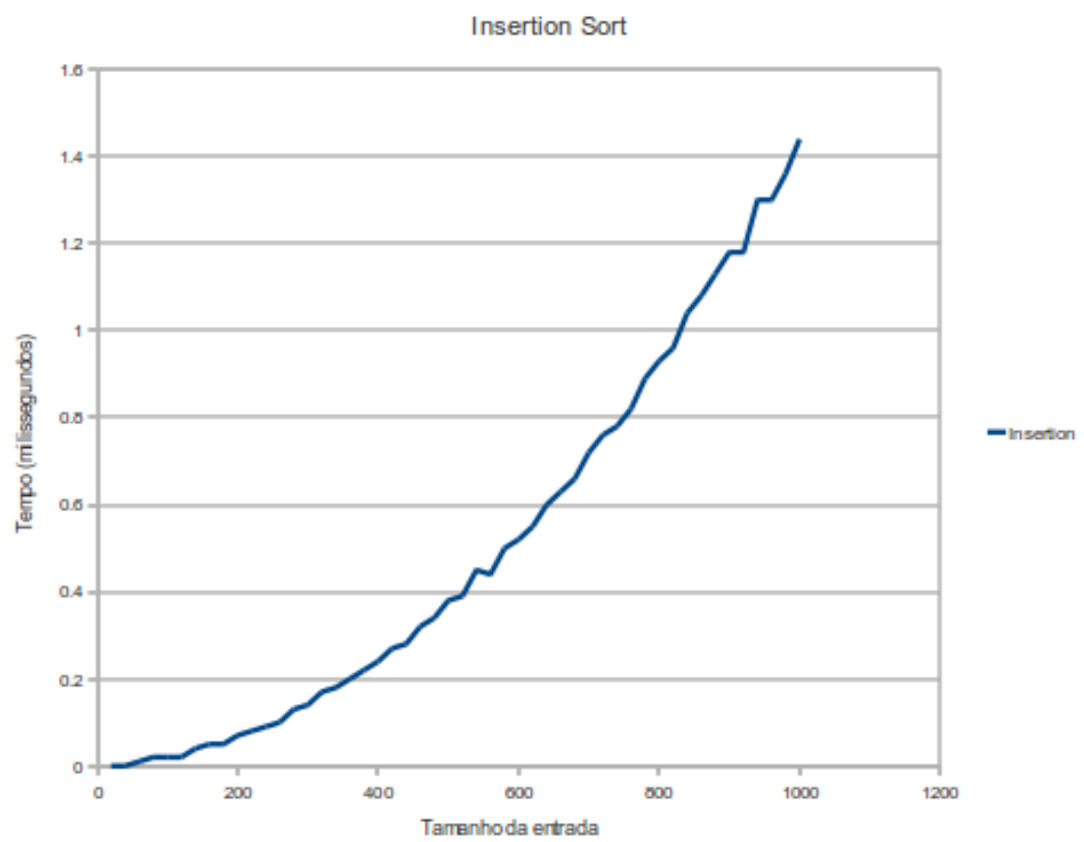


Figura 3: Gráfico de desempenho do Insertion Sort

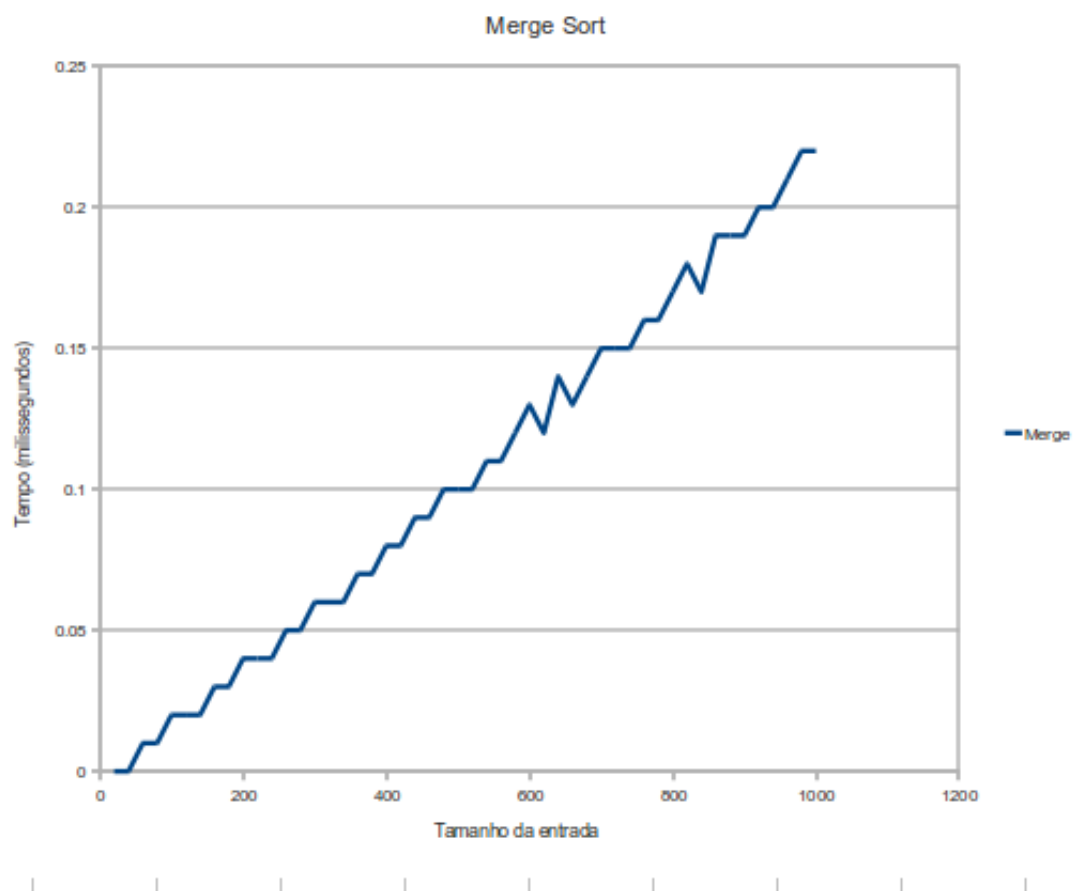


Figura 4: Gráfico de desempenho do Merge Sort

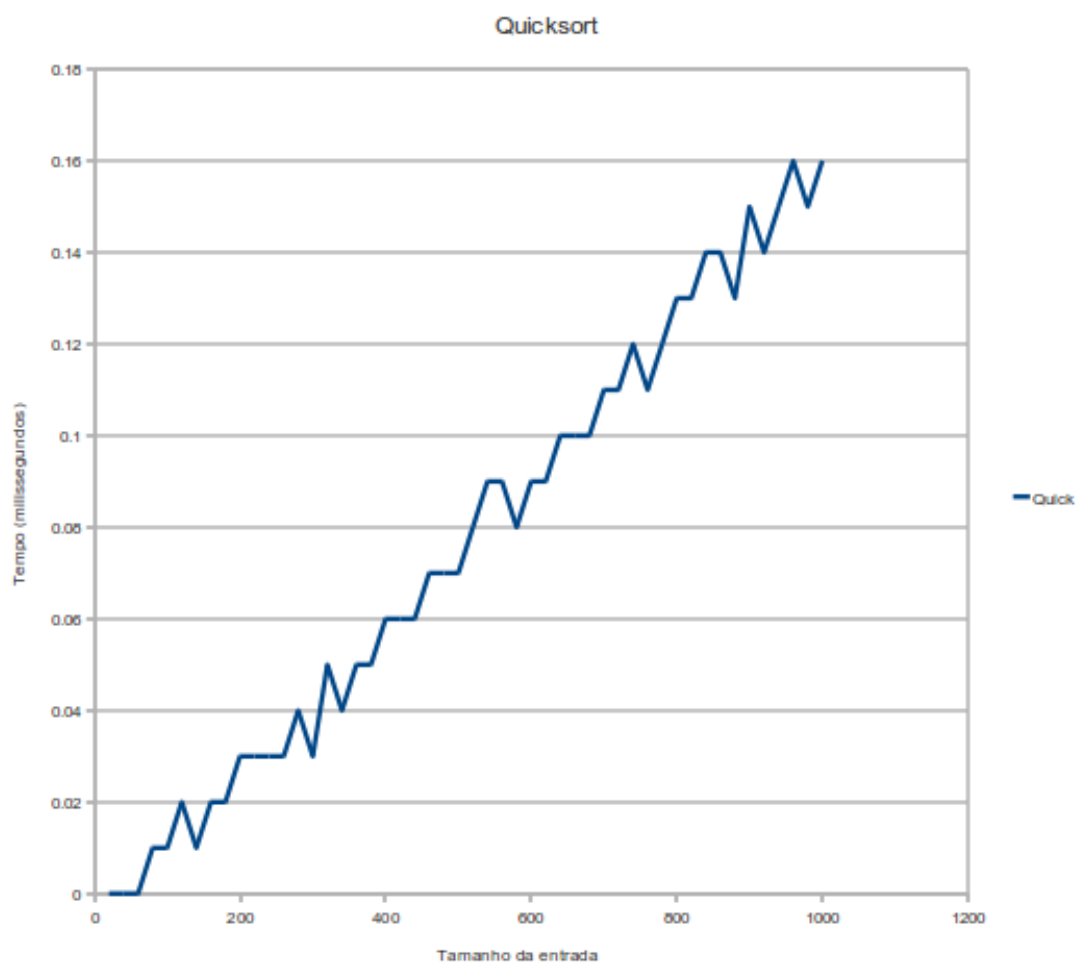


Figura 5: Gráfico de desempenho do Quicksort

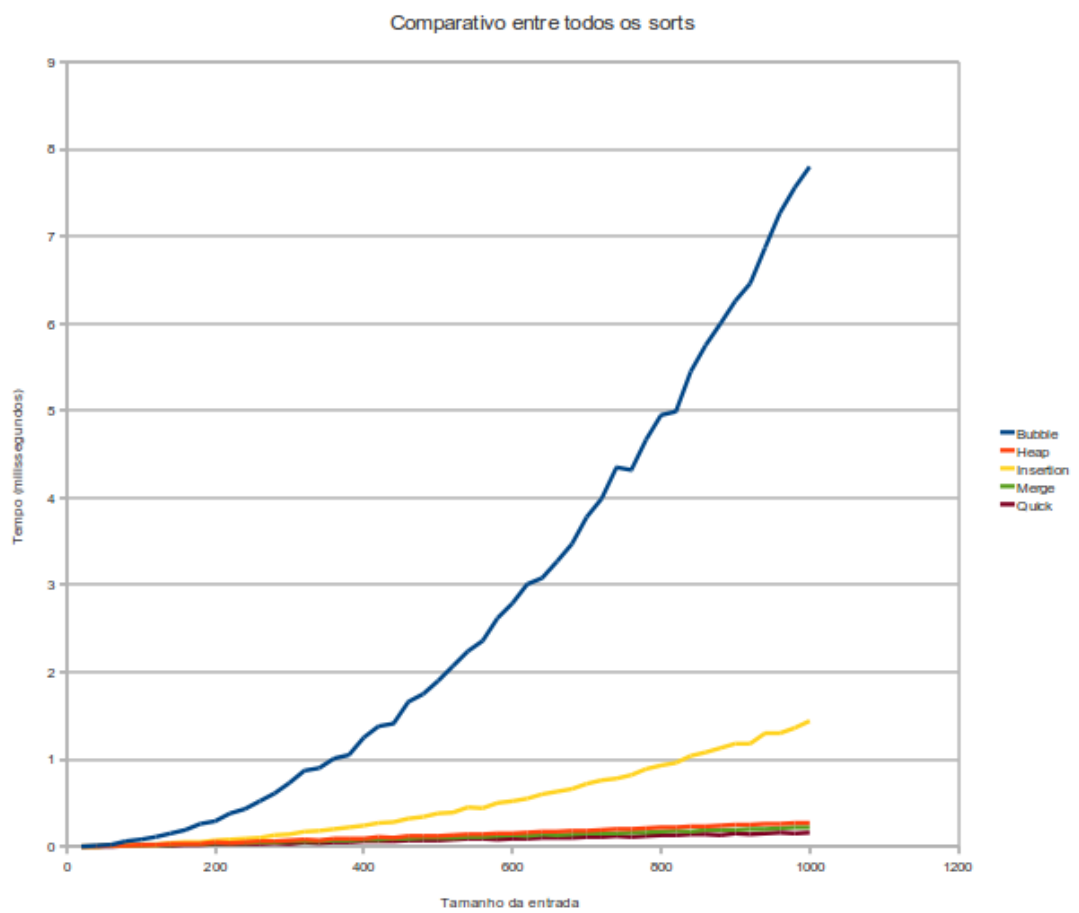


Figura 6: Gráfico comparativo de desempenho de todos os sorts