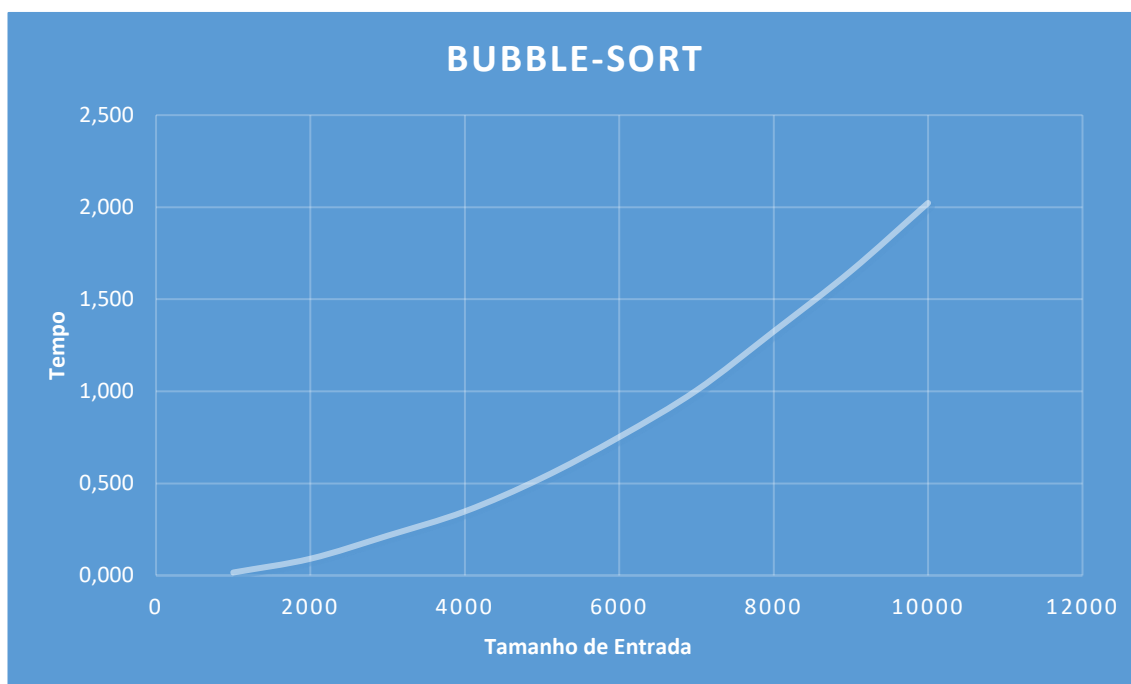
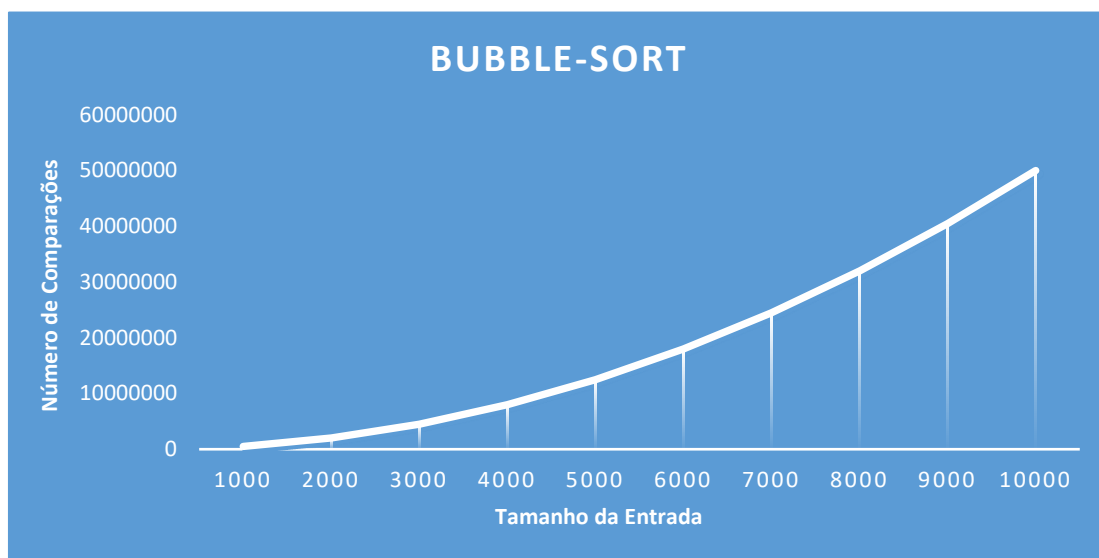


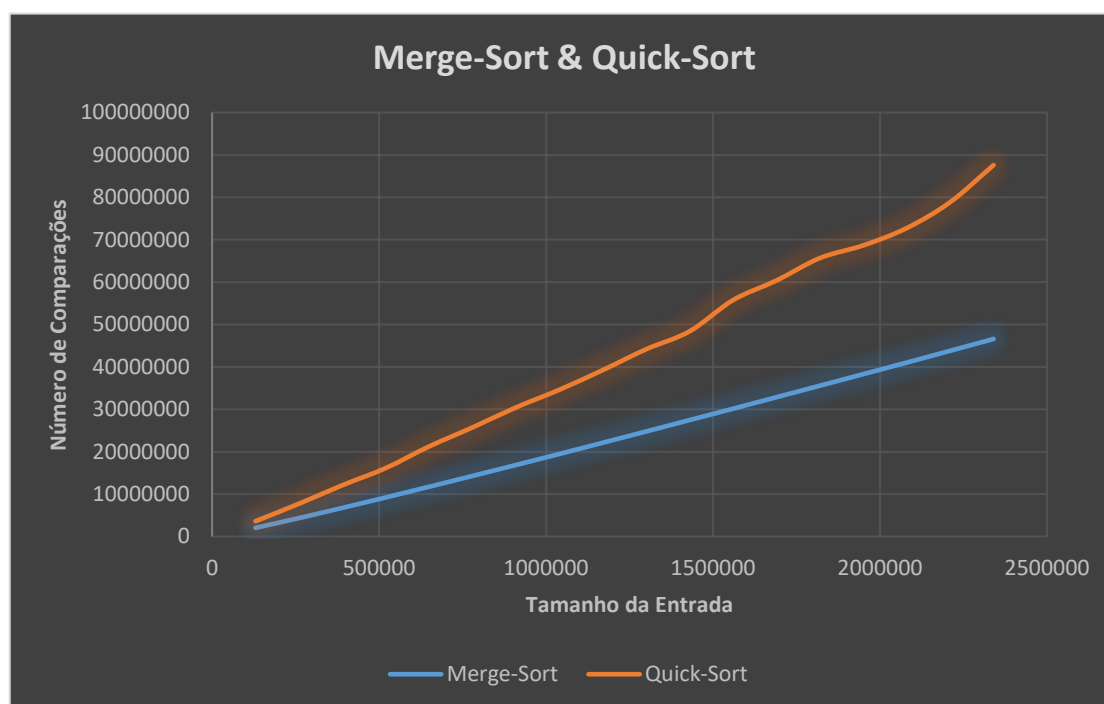
Bubble-Sort		
Tamanho da Entrada	Número de Comparações	Tempo
1000	499065	0,016
2000	1998010	0,091
3000	4498290	0,216
4000	7993247	0,348
5000	12494260	0,530
6000	17985524	0,752
7000	24494730	1,005
8000	31988497	1,325
9000	40491405	1,653
10000	49992372	2,023



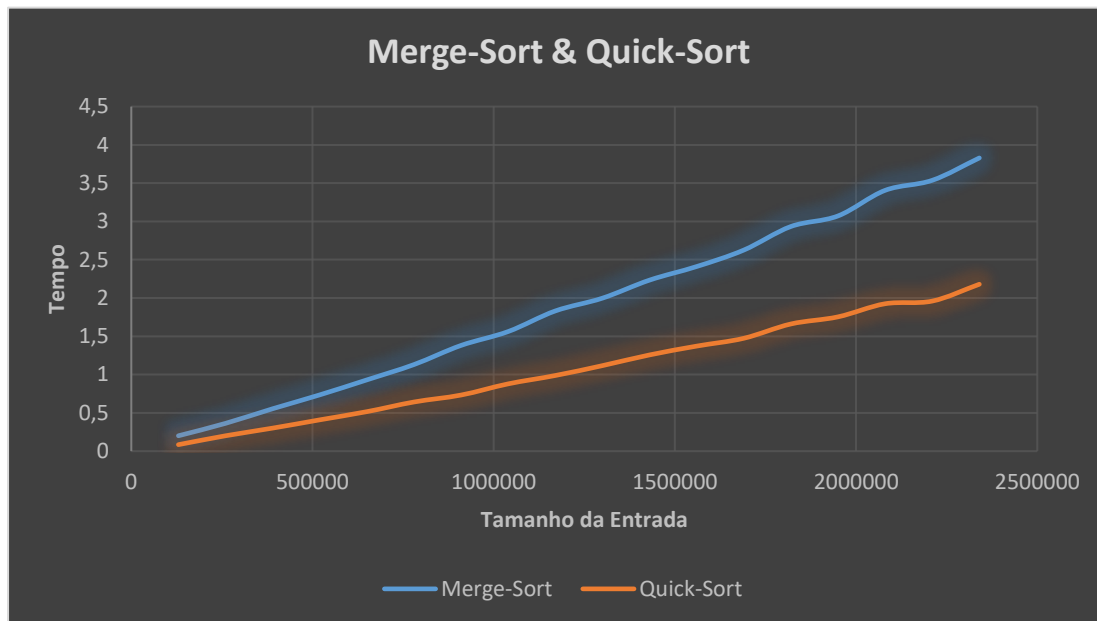
O computador atingiu 2s com  $n < 15.000$  e, conforme a orientação do professor, foram utilizados menos testes, mas foi mantido os tamanhos da entrada em múltiplos de 1000. Inclusive, note que a curva Tempo x Tamanho de Entrada tem formato de parábola, o que está de acordo com o esperado, isto é, complexidade  $O(n^2)$ .

Merge-Sort		
Tamanho da Entrada	Número de Comparações	Tempo
130000	2044086	0,201
260000	4348280	0,363
390000	6757083	0,554
520000	9217095	0,735
650000	11744148	0,931
780000	14295149	1,131
910000	16876811	1,381
1040000	19473499	1,562
1170000	22123822	1,831
1300000	24787428	1,998
1430130	27469227	2,234
1560130	30153024	2,411
1690130	32860422	2,626
1820130	35576854	2,934
1950130	38301211	3,072
2080130	41031740	3,406
2210130	43807813	3,537
2340130	46592984	3,829

Quick-Sort		
Tamanho da Entrada	Número de Comparações	Tempo
130000	3614083	0,085
260000	7732528	0,201
390000	12092791	0,301
520000	16145390	0,409
650000	21250898	0,516
780000	25736689	0,642
910000	30441160	0,733
1040000	34617078	0,879
1170000	39259275	0,988
1300000	44179982	1,117
1430130	48409252	1,257
1560130	55768541	1,372
1690130	60386712	1,474
1820130	65673572	1,661
1950130	68670752	1,754
2080130	72748549	1,925
2210130	78856086	1,960
2340130	87597003	2,180



Note que o Quick-Sort, embora mais rápido, faz mais comparações, devido a determinação da posição do pivô. Inclusive, a determinação mais irregular do pivô é refletida no gráfico, que ficou mais sinuoso do que o gráfico do Merge-Sort, que está quase uma reta (  $O(n \log n)$  ). Essa regularidade do Merge-Sort também é influenciada pela divisão sempre ao meio do vetor de elementos.



Observe que o Quick-Sort foi mais rápido do que o Merge-Sort, devido a alguns fatores, tais como: a escolha do pivô, determinando sua posição final na sequência de elementos, bem como pelo Merge-Sort escolher seu "pivô" mais "cegamente", isto é, sempre o termo do meio e, por fim, o fato do Merge-Sort trabalhar com um vetor de elementos temporário e auxiliar, o que piora a complexidade, por conta de mais instruções envolvidas, e aumenta o uso de memória.