RELATÓRIO EP3
ALGORITMOS DE ORDENAÇÃO

Algoritmo e Estrutura de Dados

MAC 0121

Feito por:
Odair Gonçalves de Oliveira
Número USP:
13671581

TESTES & VETORES

Os testes com cada algoritmo de ordenação foram feitos para calcular o número de movimentações realizadas com o vetor de palavras a ser ordenado e o número de comparações realizadas, ou seja, todas as vezes que a função "strcmp", que compara strings, foi utilizada.

Foram utilizados para os testes 5 arquivos com listas de palavras, todas com 256.000 palavras com cada palavra contendo, no máximo, 10 caracteres, minúsculos ou maiúsculos, mas sem acentos ou outras pontuações, organizados das seguintes formas:

teste_a.txt	**Palavras em ordem aleatória.**
teste_oc.txt	**Palavras organizadas em ordem crescente.**
teste_od.txt	**Palavras organizadas em ordem decrescente.**
teste_poc.txt	**Palavras organizadas em ordem crescente, mas colocando a última palavra na primeira posição.**
teste_pod.txt	**Palavras organizadas em ordem decrescente, mas colocando a última palavra na primeira posição.**

Os dados obtidos pelo teste visam explorar a relação entre as características do vetor de teste (Tanto seu tamanho, quanto sua ordem) com o desempenho de cada algoritmo, determinado pelo número de comparações e movimentações calculadas.

Assim, após análise dos dados podemos levantar comparações entre os desempenhos observados e concluir quais algoritmos são mais rápidos, estáveis e eficientes de acordo com cada teste feito.

TESTES & VETORES

Os testes foram feitos rodando a versão teste de cada algoritmo de ordenação, "nomealgoritmo_teste.c", e registrando os resultados em arquivos, disponíveis na pasta "Resultados" no arquivo entregue.

TESTES	Nº DE PALAVRAS
Teste 1	250
Teste 2	500
Teste 3	1000
Teste 4	2000
Teste 5	4000
Teste 6	8000
Teste 7	16000
Teste 8	32000
Teste 9	64000
Teste 10	128000
Teste 11	256000

Veja a seguir a estrutura dos gráficos e tabelas utilizados para registrar os resultados dos testes foram realizados.

RESULTADOS
TABELAS & GRÁFICOS

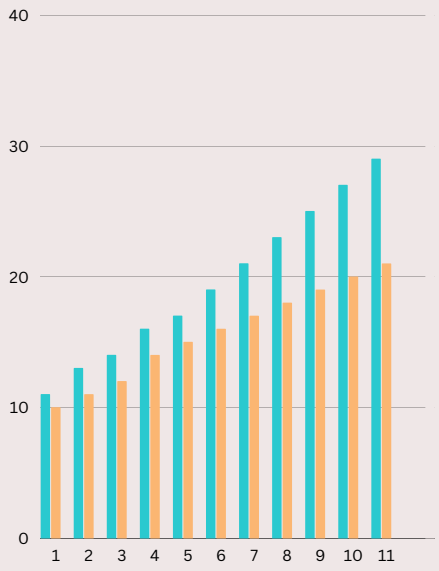
Os resultados serão expostos em tabelas e gráficos da seguinte forma:

Testes	Comparações	Trocas
Teste 1	1	4
Teste 2	2	5
Teste 3	3	6

Tabelas com os valores das movimentações / trocas e comparações realizadas em cada um dos 11 testes para cada vetor de teste testado em cada algoritmo de ordenação implementado.

Arquivo	Tempo(s)
teste_a	1,623
teste_oc	67,957
teste_od	45,646
teste_poc	67,215
teste_pod	47,417

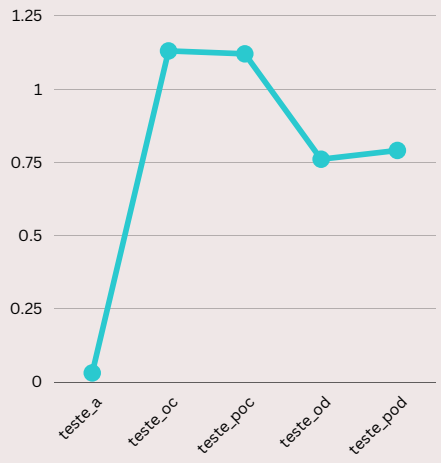
Tabelas com os valores do tempo de "real" obtido rodando o comando time no terminal com a versão teste de cada um dos algoritmos implementados. O tempo está expresso em segundos, porém nos gráficos, o tempo foi convertido para minutos para melhor observação dos resultados.



Os gráficos foram feitos utilizando os valores das tabelas calculados em logaritmo da base 2 para avaliar melhor a relação entre qual o efeito na contagem do número de movimentações e comparações ao dobrar o número de palavras a cada teste.

Para exemplificar: No gráfico ao lado vemos que o número de troca (amarelo) dobra a cada teste enquanto o número de comparações (azul) quadruplica. (Um crescimento de 1 indica uma duplicação, um crescimento de 2 indica uma quadruplicação e assim vai exponencialmente).

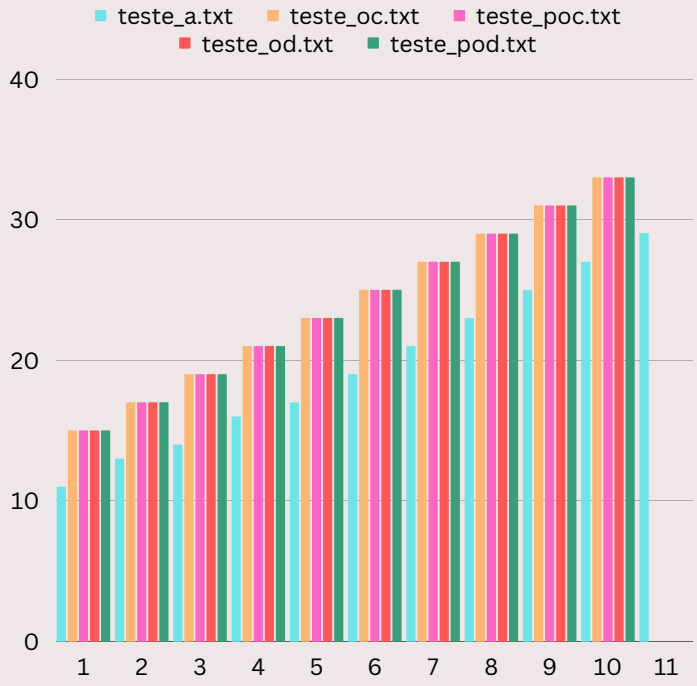
RESULTADOS
TABELAS & GRÁFICOS



Já o gráfico ao lado indica o tempo real em minutos para a execução total dos 11 (ou 10 testes*) para cada arquivo testado em cada algoritmo de ordenação.

Tal tempo foi obtido pegando o tempo "real" rodando a versão teste de cada algoritmo junto com o comando time no terminal.

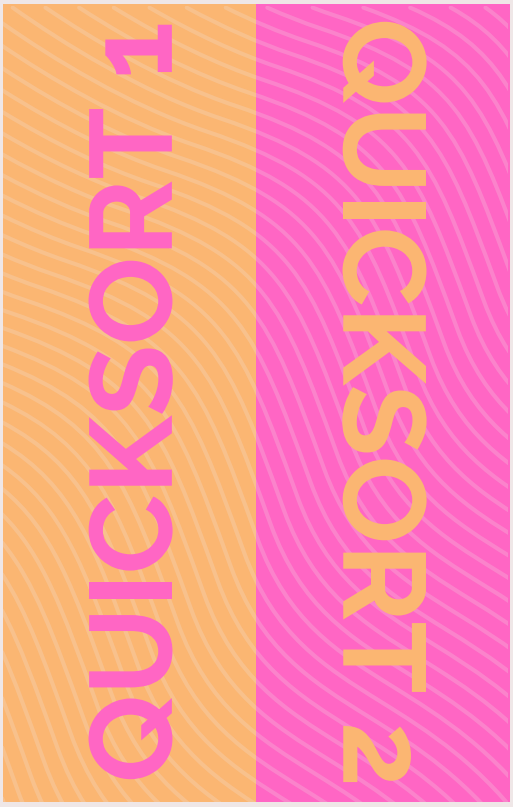
*Por questões computacionais, o teste 11 com 256 mil palavras para todos os vetores, exceto o vetor "teste_a.txt", não foi computado para as duas versões do algoritmo Quicksort.



Já o gráfico ao lado indica o número de comparações ou de movimentações (há os dois tipos) indicando o valor em LOG2(N) com N sendo o número de movimentações ou comparações obtidos em cada teste para cada um dos vetores de teste utilizados.

Por fim, encontram-se outros gráficos comparando o desempenho de todos os algoritmos implementados a partir dos dados obitos expressos nas tabelas e gráficos anteriores e uma breve conclusão sobre os resultados observados.

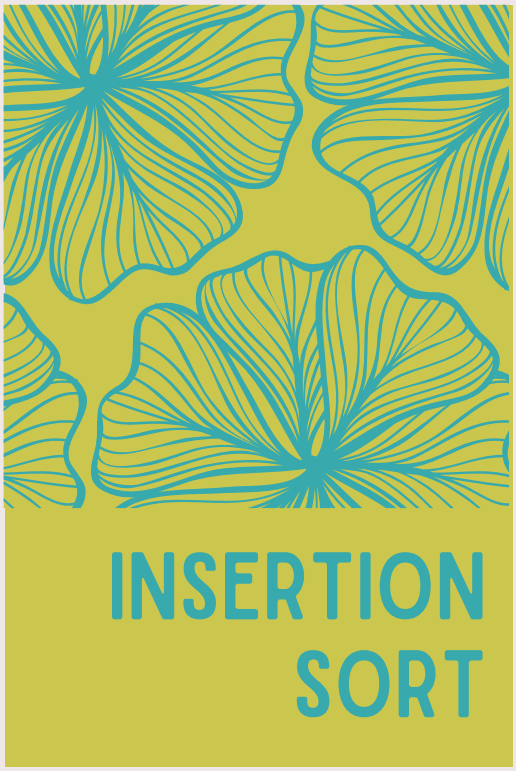
Algoritmos testados:



Páginas 7 - 10.

Páginas 11 - 14.

Páginas 15 - 18.



Páginas 19 - 22.

Páginas 23 - 26.

QUICKSORT
Versão 1

TABELAS

TESTES COM TESTE_A.TXT

Testes	Comparações	Trocas
Teste 1	2047	984
Teste 2	5807	2295
Teste 3	17350	4951
Teste 4	49402	11776
Teste 5	158782	26949
Teste 6	548224	67534
Teste 7	1978282	116350
Teste 8	7253037	240656
Teste 9	28588606	656002
Teste 10	113113427	1276652
Teste 11	454889884	2285586

TESTES COM TESTE_OC.TXT

Testes	Comparações	Trocas
Teste 1	31125	31374
Teste 2	124750	125249
Teste 3	499500	500499
Teste 4	1999000	2000999
Teste 5	7998000	8001999
Teste 6	31996000	32003999
Teste 7	127992000	128007999
Teste 8	511984000	512015999
Teste 9	2047968000	2048031999
Teste 10	8191936000	8192063999
Teste 11	-	-

TESTES COM TESTE_OD.TXT

Testes	Comparações	Trocas
Teste 1	30969	15666
Teste 2	123844	62291
Teste 3	497094	249291
Teste 4	1993594	998291
Teste 5	7986594	3996291
Teste 6	31972594	15992291
Teste 7	127944594	63984291
Teste 8	511888594	255968291
Teste 9	2047776594	1023936291
Teste 10	8191552594	4095872291
Teste 11	-	-

TESTES COM TESTE_POC.TXT

Testes	Comparações	Trocas
Teste 1	30877	31124
Teste 2	124252	124749
Teste 3	498502	499499
Teste 4	1997002	1998999
Teste 5	7994002	7997999
Teste 6	31988002	31995999
Teste 7	127976002	127991999
Teste 8	511952002	511983999
Teste 9	2047904002	2047967999
Teste 10	8191808002	8191935999
Teste 11	-	-

TESTES COM TESTE_POD.TXT

Testes	Comparações	Trocas
Teste 1	30724	15543
Teste 2	123349	62043
Teste 3	496099	248793
Teste 4	1991599	997293
Teste 5	7982599	3994293
Teste 6	31964599	15988293
Teste 7	127928599	63976293
Teste 8	511856599	255952293
Teste 9	2047712599	1023904293
Teste 10	8191424599	4095808293
Teste 11	-	-

DESEMPENHO EM TEMPO REAL:

Arquivo	Tempo(s)
teste_a	1,623
teste_oc	67,957
teste_od	45,646
teste_poc	67,215
teste_pod	47,417

QUICKSORT
Versão 1

GRÁFICOS

Legenda:
- COMPARAÇÕES;
- MOVIMENTAÇÕES / TROCAS;

GRÁFICO 1
TESTE_A.TXT

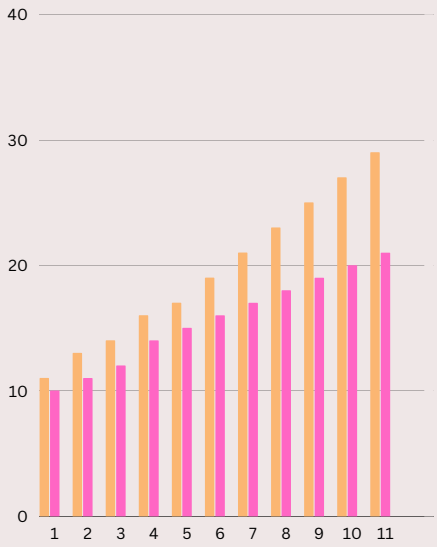


GRÁFICO
DO TEMPO

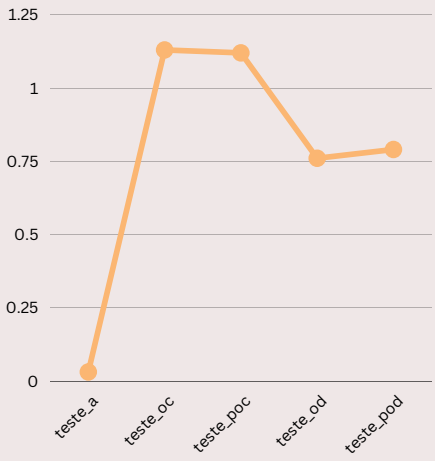


GRÁFICO 2
TESTE_OC.TXT

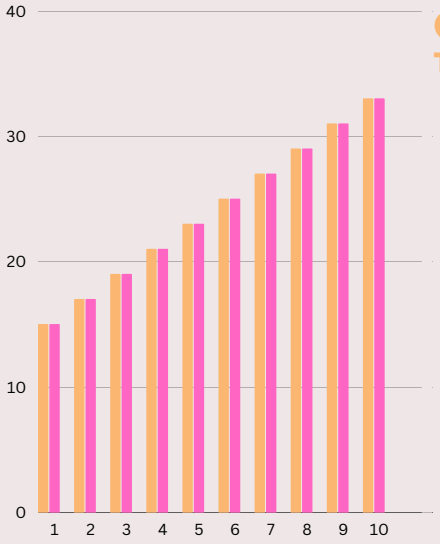


GRÁFICO 3
TESTE_POC.TXT

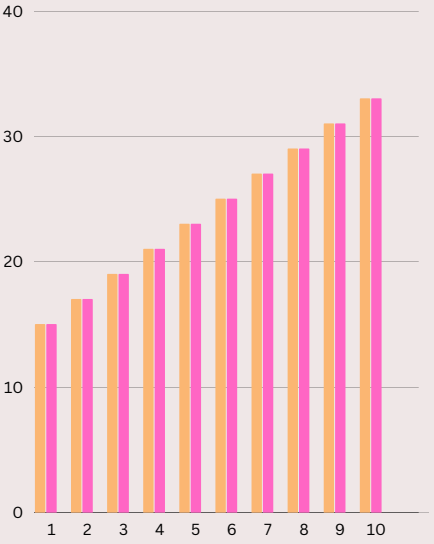


GRÁFICO 4
TESTE_OD.TXT

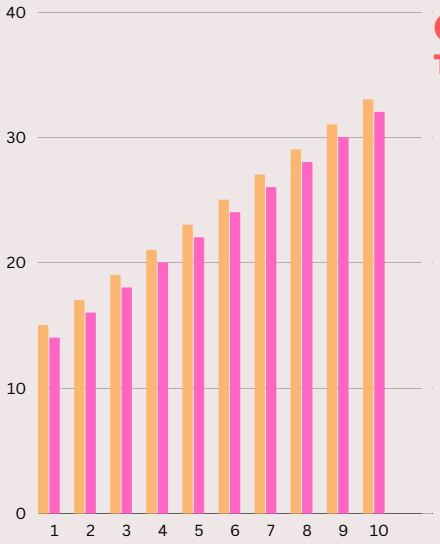
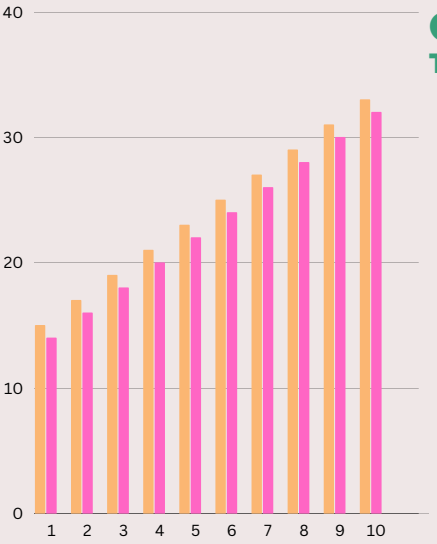
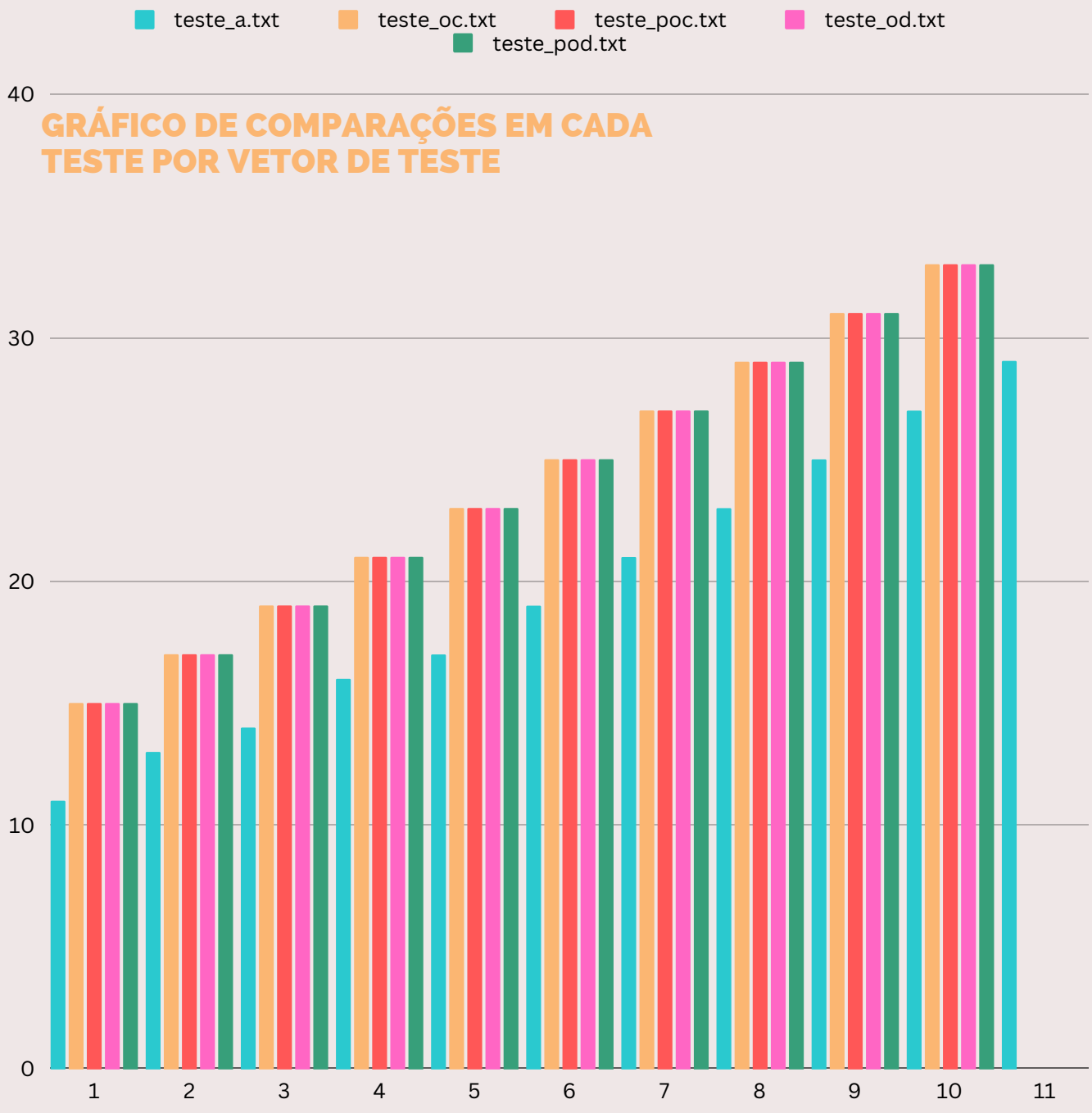


GRÁFICO 5
TESTE_POD.TXT



QUICKSORT
Versão 1

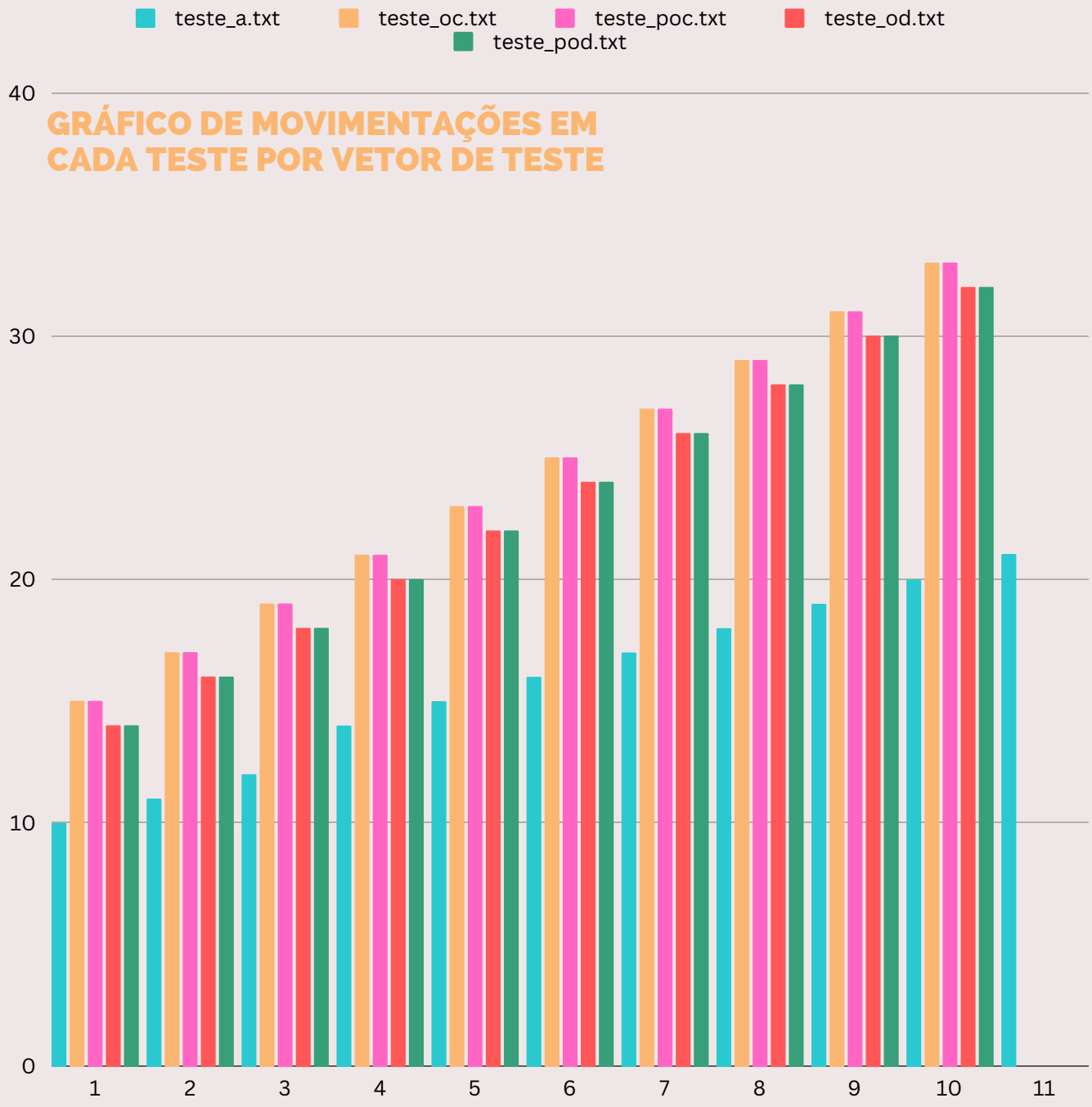
GRÁFICOS



Olhando a quantidade de comparações, os piores casos para o Quicksort 1 foram aqueles em que o vetor de teste estava ordenado de alguma forma, sem grandes diferenças para o vetores ordenados de forma crescente ou decrescente. A cada vez que o número de palavras duplicava, o número de comparações quadruplicava em todos os vetores testados.

QUICKSORT
Versão 1

GRÁFICOS



Olhando a quantidade de movimentações, os piores casos para o Quicksort 1 foram aqueles em que o vetor estava ordenado de forma crescente. O número de movimentações quadruplicou em quase todos os vetores de teste, exceto o aleatório no qual o valor apenas duplicou a cada vez que dobrava-se o número de palavras.

Comparando os vetores ordenados de forma crescente e decrescente, nos testes com os vetores decrescentes, o número de movimentações em relação aos crescentes foi aproximadamente 50% menor.

QUICKSORT
Versão 2

TABELAS

TESTES COM TESTE_A.TXT

Testes	Comparações	Trocas
Teste 1	2441	500
Teste 2	5644	1142
Teste 3	11351	2709
Teste 4	26285	6082
Teste 5	54870	13744
Teste 6	125387	30602
Teste 7	254821	67779
Teste 8	533646	149173
Teste 9	1210242	325436
Teste 10	2473091	706969
Teste 11	5253209	1525566

TESTES COM TESTE_OC.TXT

Testes	Comparações	Trocas
Teste 1	31623	249
Teste 2	125748	499
Teste 3	501498	999
Teste 4	2002998	1999
Teste 5	8005998	3999
Teste 6	32011998	7999
Teste 7	128023998	15999
Teste 8	512047998	31999
Teste 9	2048095998	63999
Teste 10	8192191998	127999
Teste 11	-	-

TESTES COM TESTE_OD.TXT

Testes	Comparações	Trocas
Teste 1	31186	249
Teste 2	123686	499
Teste 3	496186	999
Teste 4	1991186	1999
Teste 5	7981186	3999
Teste 6	31961186	7999
Teste 7	127921186	15999
Teste 8	511841186	31999
Teste 9	2047681186	63999
Teste 10	8191361186	127999
Teste 11	-	-

TESTES COM TESTE_POC.TXT

Testes	Comparações	Trocas
Teste 1	31374	249
Teste 2	125249	499
Teste 3	500499	999
Teste 4	2000999	1999
Teste 5	8001999	3999
Teste 6	32003999	7999
Teste 7	128007999	15999
Teste 8	512015999	31999
Teste 9	2048031999	63999
Teste 10	8192063999	127999
Teste 11	-	-

TESTES COM TESTE_POD.TXT

Testes	Comparações	Trocas
Teste 1	31193	249
Teste 2	123693	499
Teste 3	496193	999
Teste 4	1991193	1999
Teste 5	7981193	3999
Teste 6	31961193	7999
Teste 7	127921193	15999
Teste 8	511841193	31999
Teste 9	2047681193	63999
Teste 10	8191361193	127999
Teste 11	-	-

DESEMPENHO EM TEMPO REAL:

Arquivo	Tempo(s)
teste_a	0,210
teste_oc	51,036
teste_od	50,909
teste_poc	50,947
teste_pod	51,212

QUICKSORT
Versão 2

GRÁFICOS

Legenda:
- **COMPARAÇÕES;**
- **MOVIMENTAÇÕES / TROCAS;**

GRÁFICO 1
TESTE_A.TXT

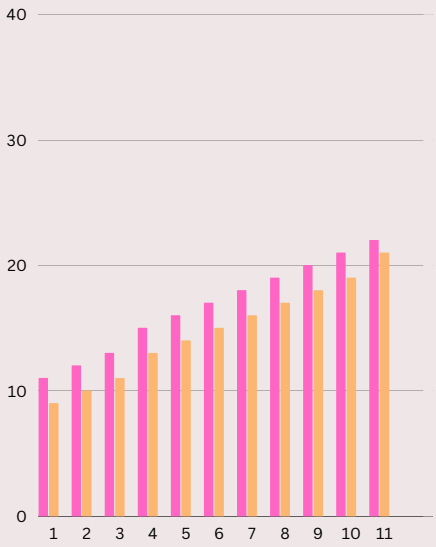


GRÁFICO
DO TEMPO

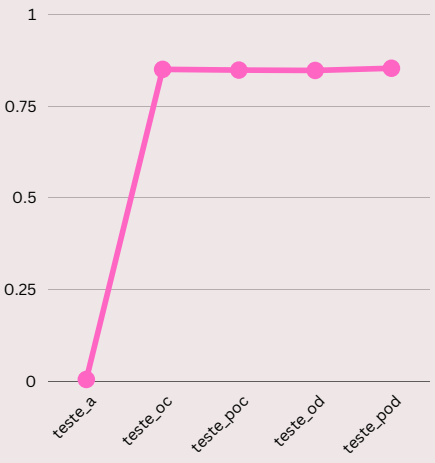


GRÁFICO 2
TESTE_OC.TXT

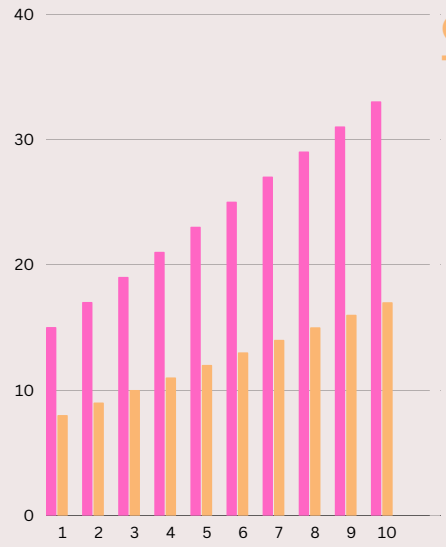


GRÁFICO 3
TESTE_POC.TXT

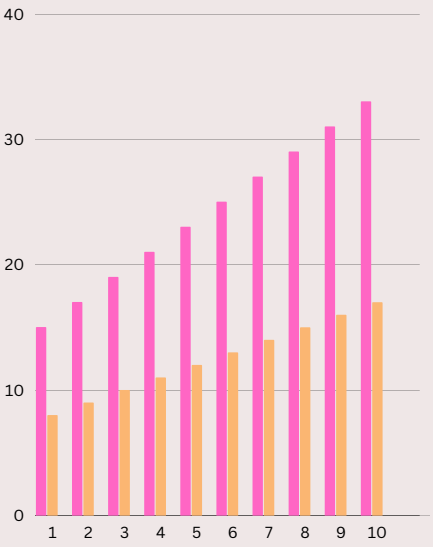


GRÁFICO 4
TESTE_OD.TXT

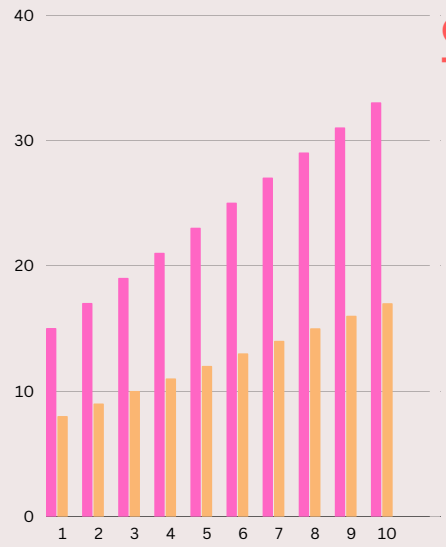
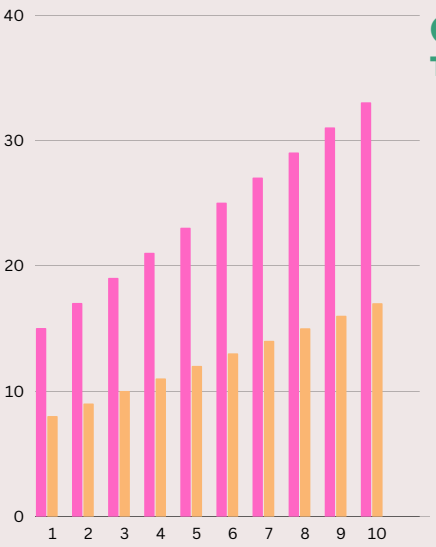
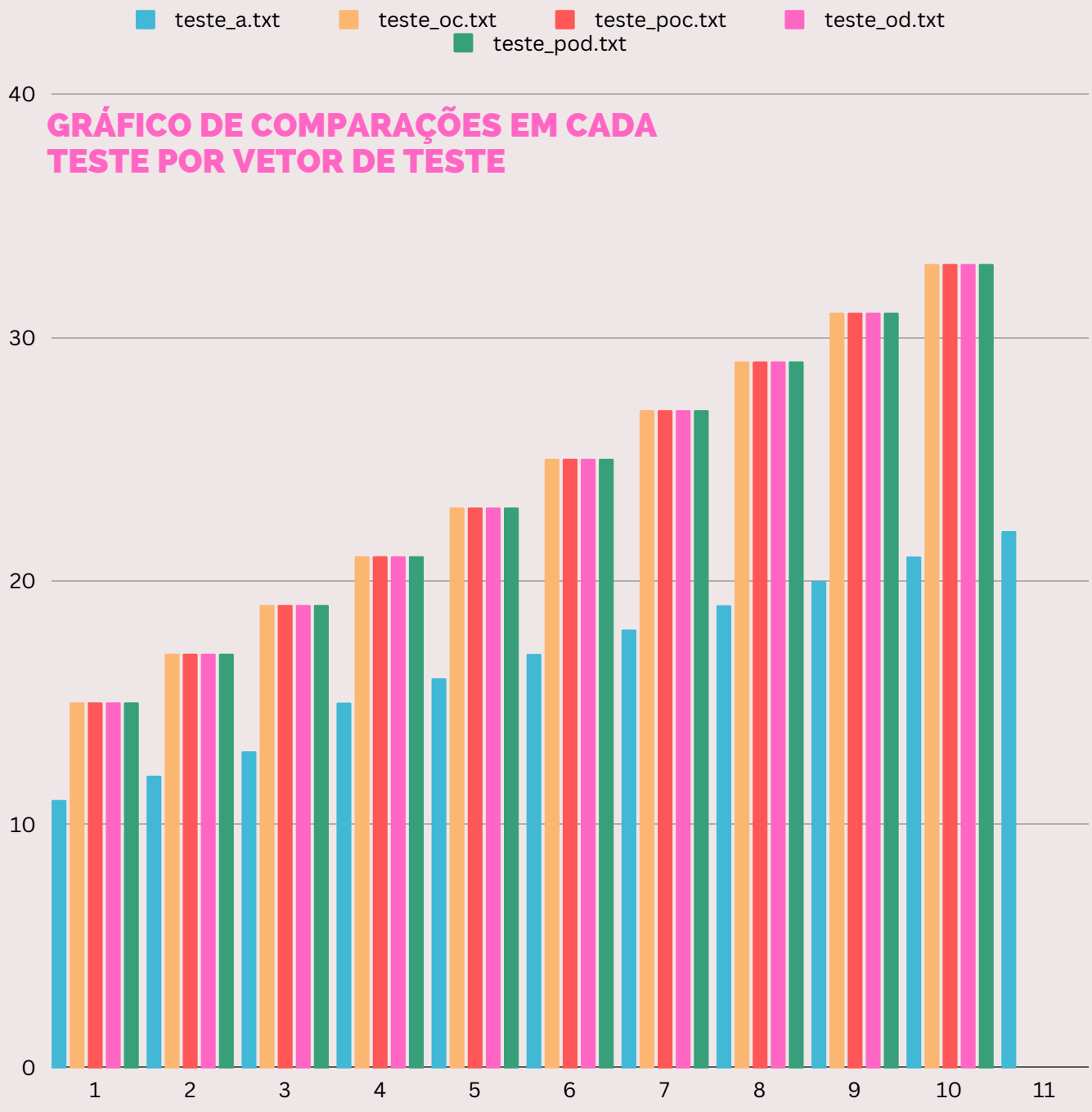


GRÁFICO 5
TESTE_POD.TXT

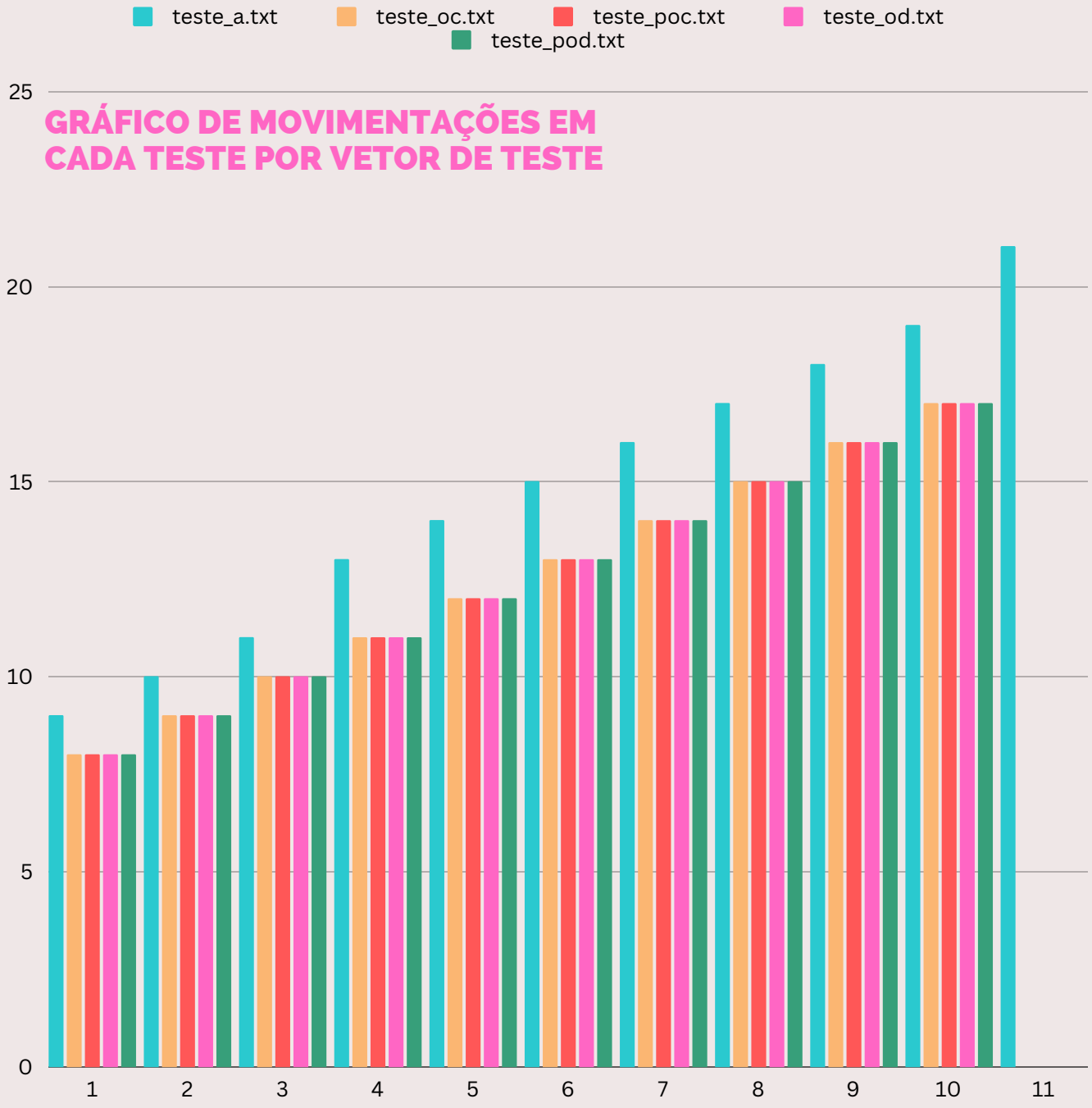


GRÁFICOS



Olhando a quantidade de comparações, os piores casos para o Quicksort 2 foram aqueles em que o vetor de teste estava ordenado de alguma forma, sem grandes diferenças para o vetores ordenados de forma crescente ou decrescente. A cada vez que o número de palavras duplicava, o número de comparações quadruplicava em todos os vetores testados, exceto no aleatório, no qual o número de comparações apenas duplicou.

GRÁFICOS



Olhando a quantidade de movimentações, o pior caso para o Quicksort 2 foi o teste com o vetor aleatório. O número de movimentações apenas duplicou em todos os vetores de teste.

Nos vetores ordenados, o número de movimentações foi praticamente igual, muito próximo da quantidade N de palavras de cada teste. Mas, mesmo com número maior de movimentações, o melhor caso, comparando o tempo, foi com o vetor aleatório.

TABELAS

TESTES COM TESTE_A.TXT

Testes	Comparações	Trocas
Teste 1	3862	1806
Teste 2	8598	4049
Teste 3	19216	9108
Teste 4	42186	20093
Teste 5	92222	44111
Teste 6	200460	96230
Teste 7	432474	208237
Teste 8	928166	448083
Teste 9	1982428	959214
Teste 10	4219844	2045922
Teste 11	8936298	4340149

TESTES COM TESTE_OC.TXT

Testes	Comparações	Trocas
Teste 1	4106	1928
Teste 2	9208	4354
Teste 3	20416	9708
Teste 4	44600	21300
Teste 5	98284	47142
Teste 6	213276	102638
Teste 7	456760	220380
Teste 8	975048	471524
Teste 9	2081336	1008668
Teste 10	4431212	2151606
Teste 11	9354756	4549378

TESTES COM TESTE_OD.TXT

Testes	Comparações	Trocas
Teste 1	3426	1588
Teste 2	7858	3679
Teste 3	17650	8325
Teste 4	39422	18711
Teste 5	86862	41431
Teste 6	189554	90777
Teste 7	411746	197873
Teste 8	889010	428505
Teste 9	1899074	917537
Teste 10	4055014	1963507
Teste 11	8614354	4179177

TESTES COM TESTE_POC.TXT

Testes	Comparações	Trocas
Teste 1	4080	1915
Teste 2	9182	4341
Teste 3	20378	9689
Teste 4	44810	21405
Teste 5	97466	46733
Teste 6	212526	102263
Teste 7	458534	221267
Teste 8	977490	472745
Teste 9	2078086	1007043
Teste 10	4418610	2145305
Teste 11	9374370	4559185

TESTES COM TESTE_POD.TXT

Testes	Comparações	Trocas
Teste 1	3456	1603
Teste 2	7884	3692
Teste 3	17740	8370
Teste 4	39472	18736
Teste 5	86712	41356
Teste 6	189880	90940
Teste 7	412648	198324
Teste 8	885504	426752
Teste 9	1898836	917418
Teste 10	4052252	1962126
Teste 11	8641916	4192958

DESEMPENHO EM TEMPO REAL:

Arquivo	Tempo(s)
teste_a	0,211
teste_oc	0,222
teste_od	0,213
teste_poc	0,192
teste_pod	0,229

HEAPSORT
Versão Única

GRÁFICOS

Legenda:
- **COMPARAÇÕES;**
- **MOVIMENTAÇÕES / TROCAS;**

GRÁFICO 1
TESTE_A.TXT

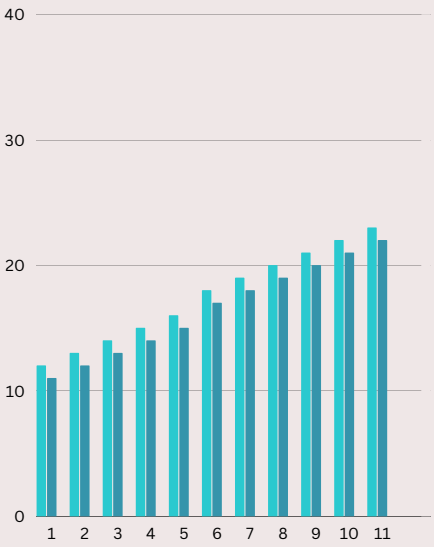


GRÁFICO DO TEMPO

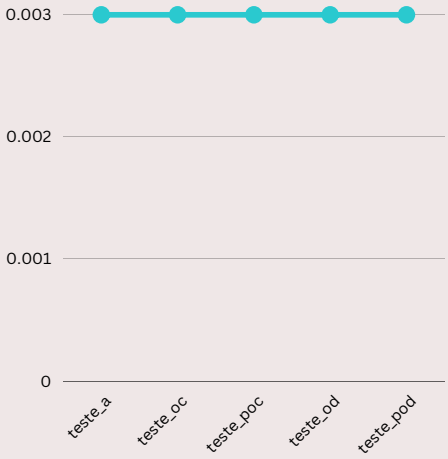


GRÁFICO 2
TESTE_OC.TXT

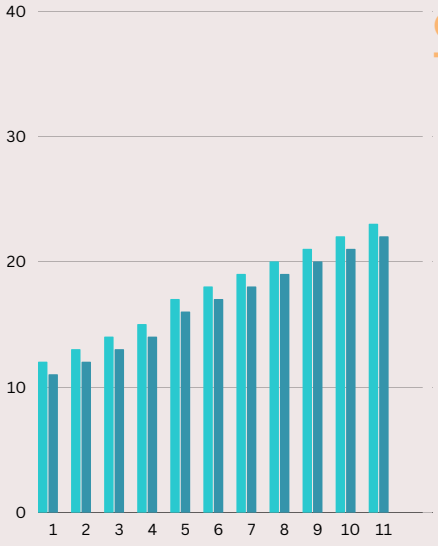


GRÁFICO 3
TESTE_POC.TXT

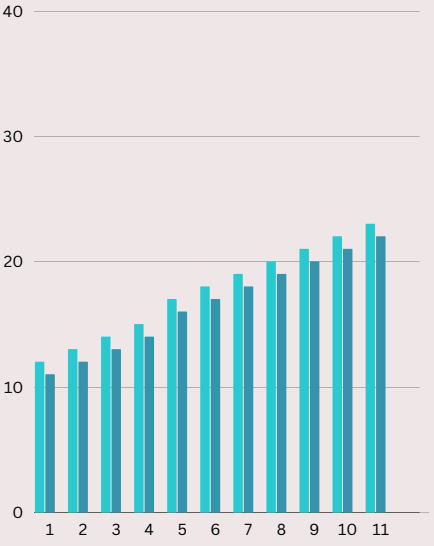


GRÁFICO 4
TESTE_OD.TXT

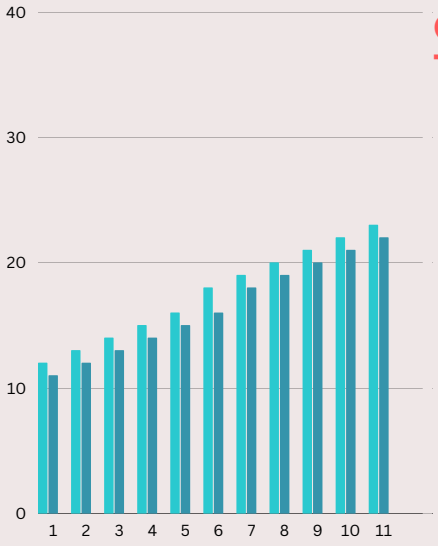
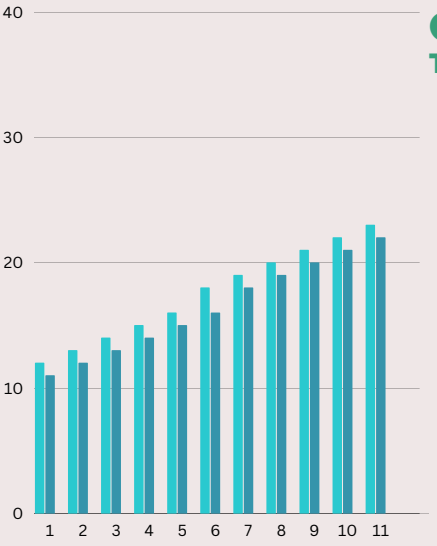
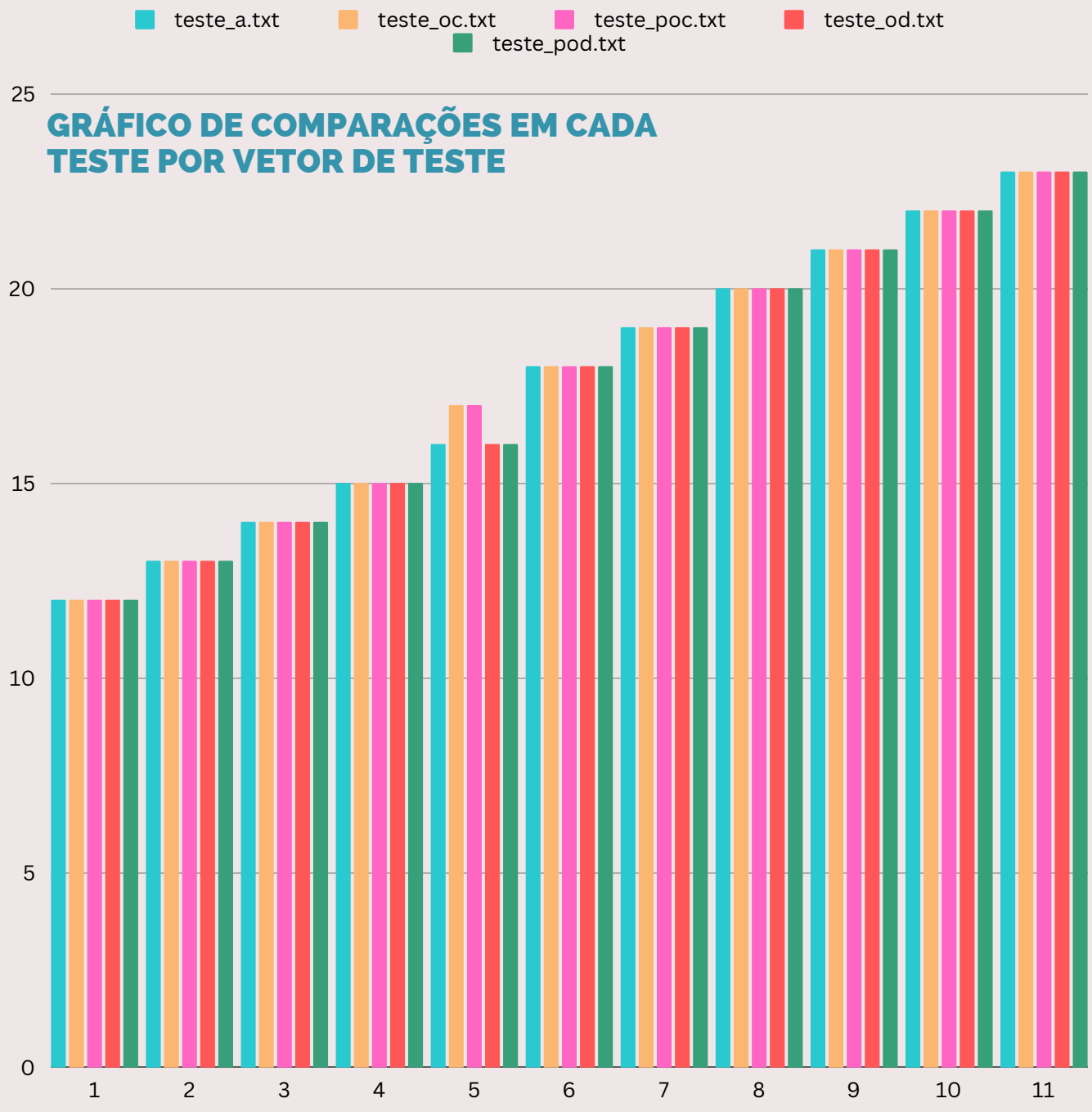


GRÁFICO 5
TESTE_POD.TXT



GRÁFICOS



Olhando a quantidade de comparações, observamos que o Heapsort se comportou de maneira bem estável, não sendo tão afetado pela ordenação dos vetores de teste. Por uma diferença pífia, os testes com maior número de comparações foram os testes com os vetores ordenados de forma crescente, mas sem diferenças notáveis, como atesta o gráfico e tabela do tempo.

Em todos os casos, o número de comparações duplicou ao se dobrar o número de palavras e foi, aproximadamente, o dobro do número de trocas realizadas.

GRÁFICOS

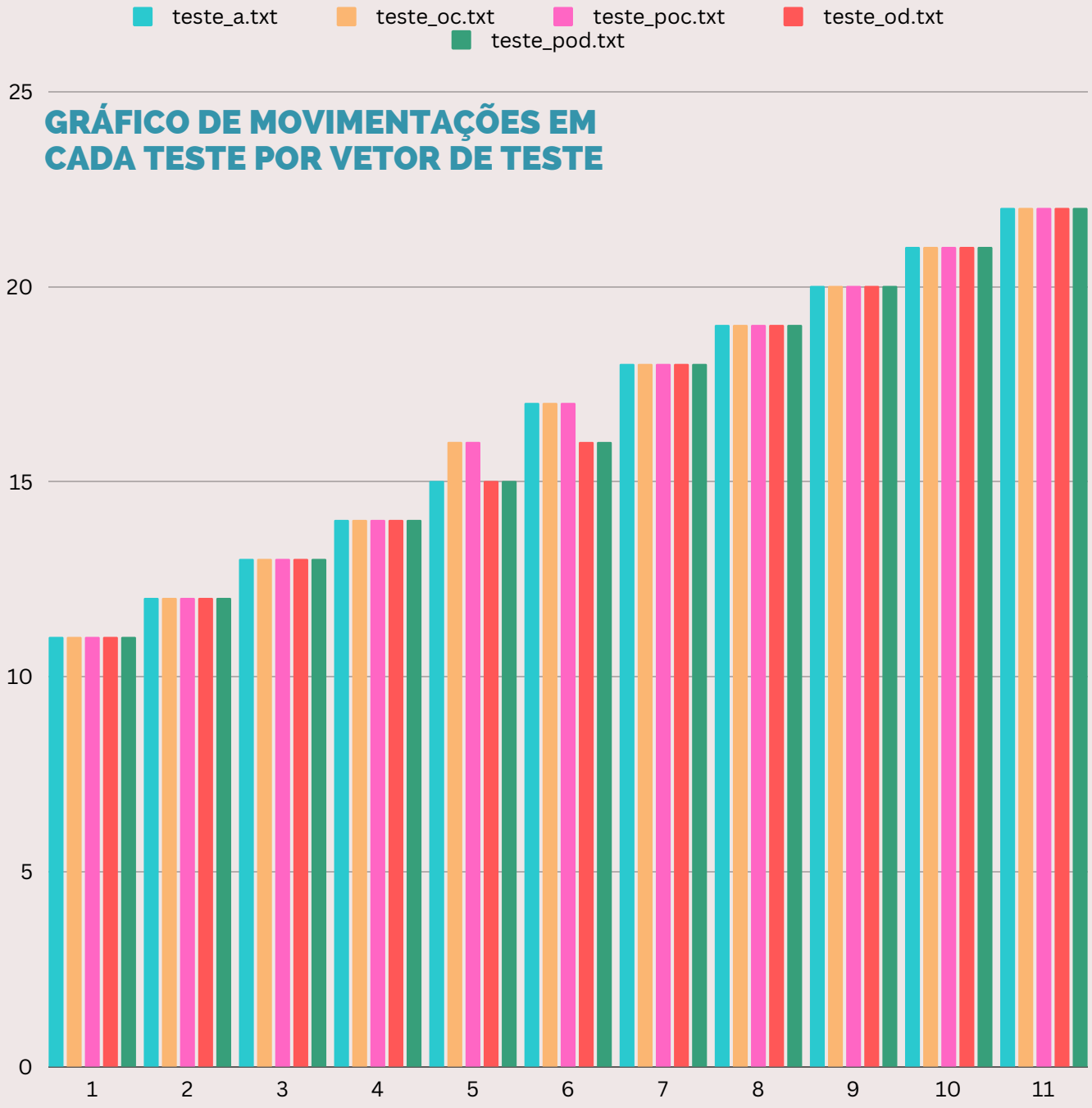


GRÁFICO DE MOVIMENTAÇÕES EM CADA TESTE POR VETOR DE TESTE

Olhando a quantidade de movimentações, observamos a mesma estabilidade que observamos no número de comparações, o que era esperado já que, no geral, a forma de orientação do vetor de teste não impactua tanto na estruturação do vetor no formato de heap binário.

Destaca-se novamente que o número de movimentações foi, aproximadamente, 50% do número de comparações em cada teste.

TABELAS

TESTES COM TESTE_A.TXT

Testes	Comparações	Trocas
Teste 1	13880	13880
Teste 2	57614	57614
Teste 3	236540	236540
Teste 4	976918	976918
Teste 5	3934837	3934837
Teste 6	15731063	15731063
Teste 7	62655235	62655235
Teste 8	253981933	253981933
Teste 9	1006967537	1006967537
Teste 10	4013119171	4013119171
Teste 11	16099135395	16099135395

TESTES COM TESTE_OC.TXT

Testes	Comparações	Trocas
Teste 1	249	249
Teste 2	499	499
Teste 3	999	999
Teste 4	1999	1999
Teste 5	3999	3999
Teste 6	7999	7999
Teste 7	15999	15999
Teste 8	31999	31999
Teste 9	63999	63999
Teste 10	127999	127999
Teste 11	255999	255999

TESTES COM TESTE_OD.TXT

Testes	Comparações	Trocas
Teste 1	31368	31368
Teste 2	125243	125243
Teste 3	500493	500493
Teste 4	2000993	2000993
Teste 5	8001993	8001993
Teste 6	32003993	32003993
Teste 7	128007993	128007993
Teste 8	512015993	512015993
Teste 9	2048031993	2048031993
Teste 10	8192063993	8192063993
Teste 11	32768127993	32768127993

TESTES COM TESTE_POC.TXT

Testes	Comparações	Trocas
Teste 1	498	498
Teste 2	998	998
Teste 3	1998	1998
Teste 4	3998	3998
Teste 5	7998	7998
Teste 6	15998	15998
Teste 7	31998	31998
Teste 8	63998	63998
Teste 9	127998	127998
Teste 10	255998	255998
Teste 11	511998	511998

TESTES COM TESTE_POD.TXT

Testes	Comparações	Trocas
Teste 1	31119	31119
Teste 2	124744	124744
Teste 3	499494	499494
Teste 4	1998994	1998994
Teste 5	7997994	7997994
Teste 6	31995994	31995994
Teste 7	127991994	127991994
Teste 8	511983994	511983994
Teste 9	2047967994	2047967994
Teste 10	8191935994	8191935994
Teste 11	32767871994	32767871994

DESEMPENHO EM TEMPO REAL:

Arquivo	Tempo(s)
teste_a	178,164
teste_oc	0,067
teste_od	364,574
teste_poc	0,077
teste_pod	356,459

INSERTION_SORT
Versão Única

GRÁFICOS

Legenda:
- **COMPARAÇÕES;**
- **MOVIMENTAÇÕES / TROCAS;**

GRÁFICO 1
TESTE_A.TXT

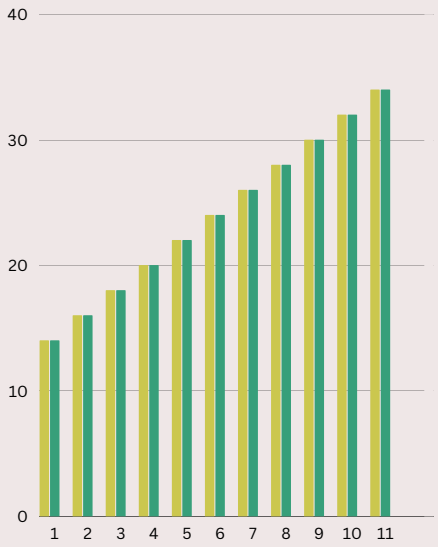


GRÁFICO
DO TEMPO

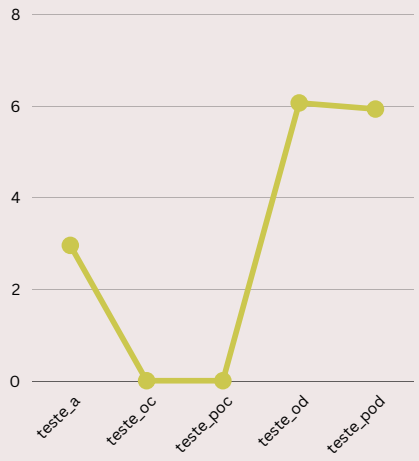


GRÁFICO 2
TESTE_OC.TXT

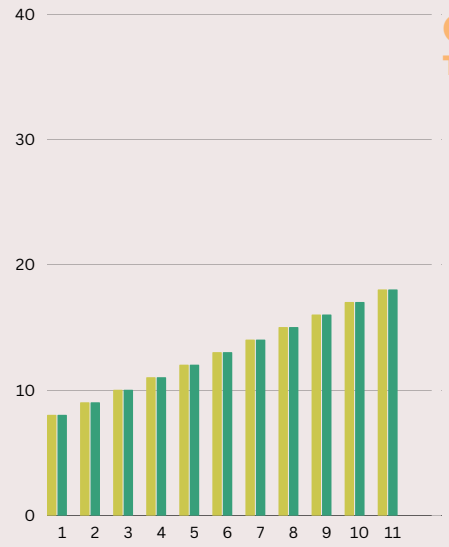


GRÁFICO 3
TESTE_POC.TXT

GRÁFICO 4
TESTE_OD.TXT

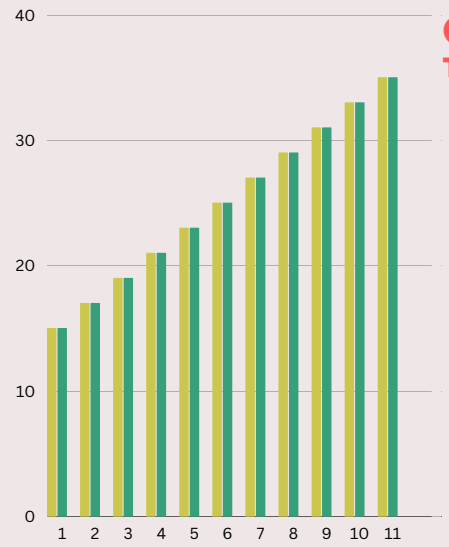
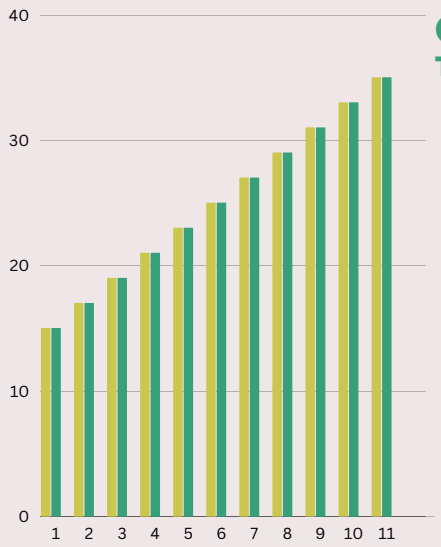
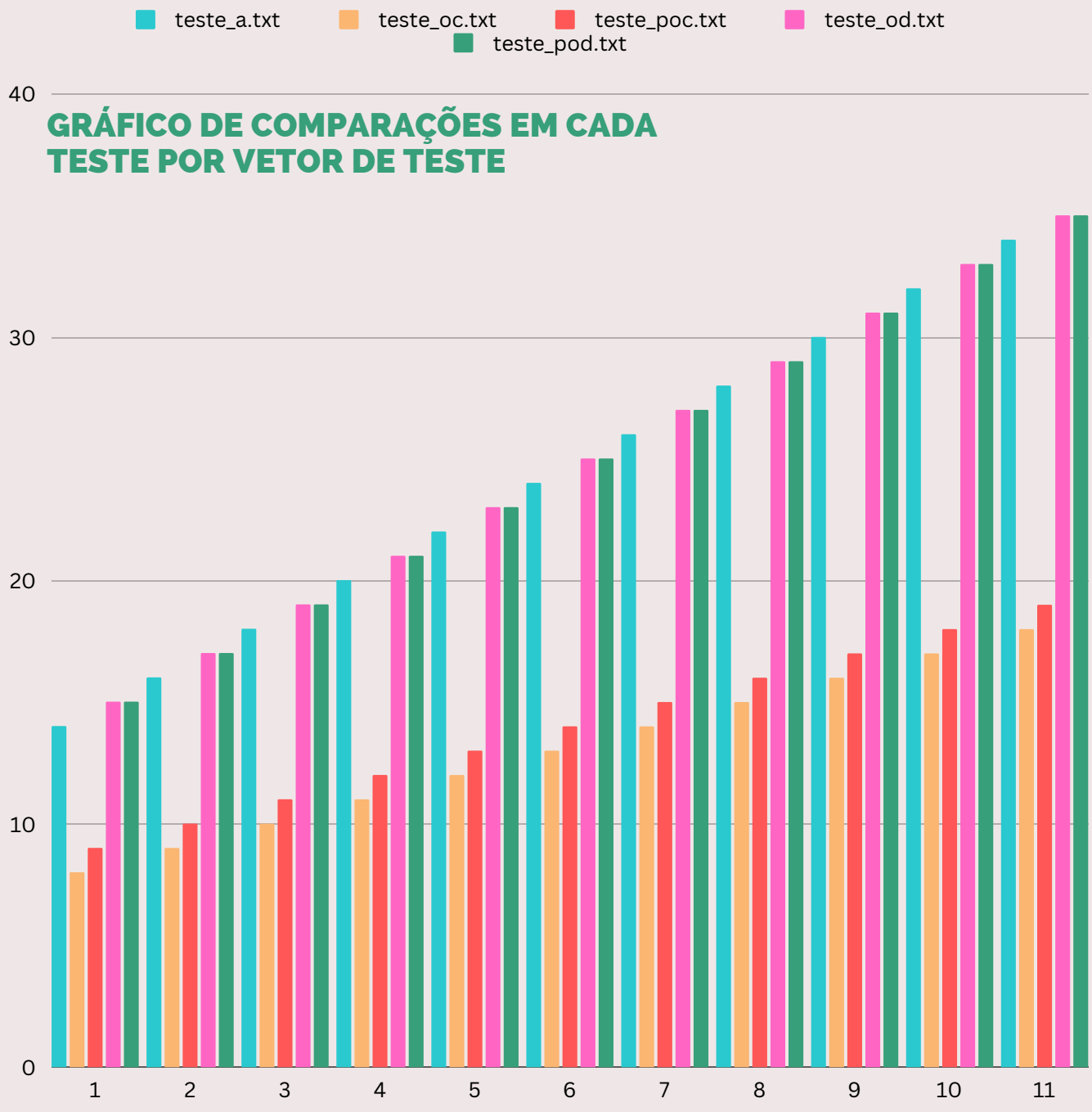


GRÁFICO 5
TESTE_POD.TXT



GRÁFICOS



Olhando a quantidade de comparações, observamos que o Insertion teve os piores desempenhos com os vetores ordenados de forma decrescente e com o aleatório. Já o melhor foi com os vetores orientados de forma crescente.

Nos piores casos, o número de comparações quadruplicou a cada teste, juntamente com o número de trocas. Já nos melhores casos, o número de comparações ficou próximo a N para o teste com "teste_oc.txt" e 2N para o teste com "teste_poc.txt".

GRÁFICOS

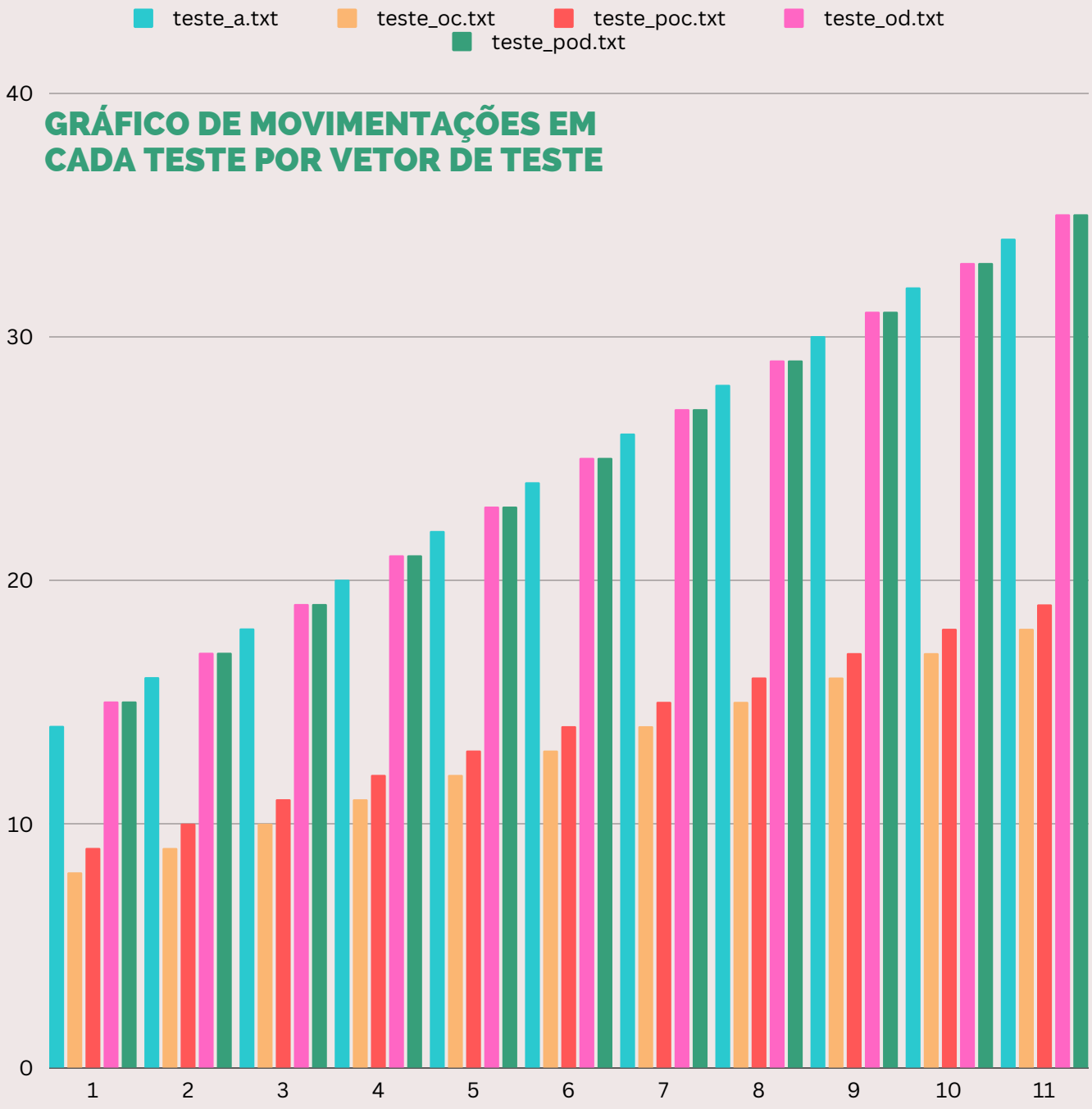


GRÁFICO DE MOVIMENTAÇÕES EM CADA TESTE POR VETOR DE TESTE

Como a quantidade de comparações é igual a quantidade de movimentações no caso do Insertion, a análise feita anteriormente é análoga à análise que seria realizada aqui.

TABELAS

TESTES COM TESTE_A.TXT

Testes	Comparações	Trocas
Teste 1	1666	1994
Teste 2	3849	4488
Teste 3	8715	9976
Teste 4	19438	21952
Teste 5	42864	47904
Teste 6	93715	103808
Teste 7	203384	223616
Teste 8	438605	479232
Teste 9	941171	1022464
Teste 10	2010300	2172928
Teste 11	4276783	4601856

TESTES COM TESTE_OC.TXT

Testes	Comparações	Trocas
Teste 1	1011	1994
Teste 2	2272	4488
Teste 3	5044	9976
Teste 4	11088	21952
Teste 5	24176	47904
Teste 6	52352	103808
Teste 7	112704	223616
Teste 8	241408	479232
Teste 9	514816	1022464
Teste 10	1093632	2172928
Teste 11	2315264	4601856

TESTES COM TESTE_OD.TXT

Testes	Comparações	Trocas
Teste 1	986	1994
Teste 2	2219	4488
Teste 3	4935	9976
Teste 4	10867	21952
Teste 5	23731	47904
Teste 6	51459	103808
Teste 7	110915	223616
Teste 8	237827	479232
Teste 9	507651	1022464
Teste 10	1079299	2172928
Teste 11	2286595	4601856

TESTES COM TESTE_POC.TXT

Testes	Comparações	Trocas
Teste 1	1252	1994
Teste 2	2762	4488
Teste 3	6033	9976
Teste 4	13076	21952
Teste 5	28163	47904
Teste 6	60338	103808
Teste 7	128689	223616
Teste 8	273392	479232
Teste 9	578799	1022464
Teste 10	1221614	2172928
Teste 11	2571245	4601856

TESTES COM TESTE_POD.TXT

Testes	Comparações	Trocas
Teste 1	994	1994
Teste 2	2228	4488
Teste 3	4945	9976
Teste 4	10878	21952
Teste 5	23743	47904
Teste 6	51472	103808
Teste 7	110929	223616
Teste 8	237842	479232
Teste 9	507667	1022464
Teste 10	1079316	2172928
Teste 11	2286613	4601856

DESEMPENHO EM TEMPO REAL:

Arquivo	Tempo(s)
teste_a	0,163
teste_oc	0,118
teste_od	0,118
teste_poc	0,12
teste_pod	0,12

MERGESORT
Versão Única

GRÁFICOS

Legenda:
- **COMPARAÇÕES;**
- **MOVIMENTAÇÕES / TROCAS;**

GRÁFICO 1
TESTE_A.TXT

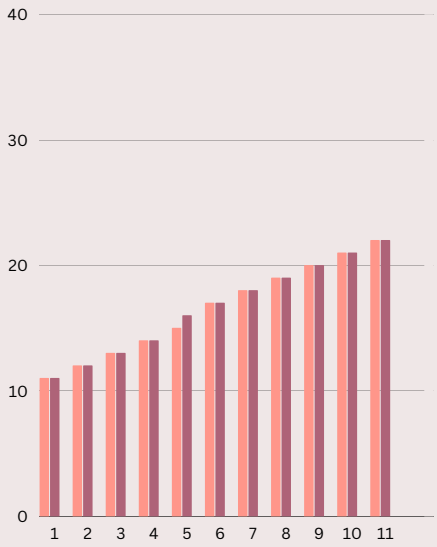


GRÁFICO DO TEMPO

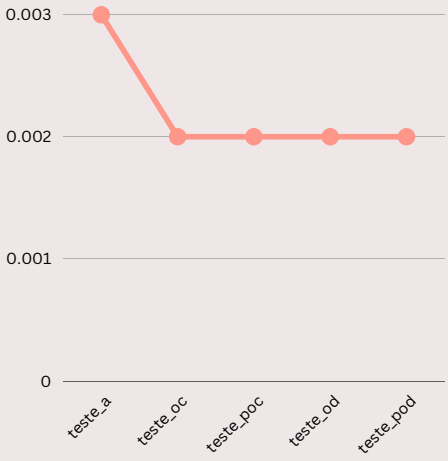


GRÁFICO 2
TESTE_OC.TXT

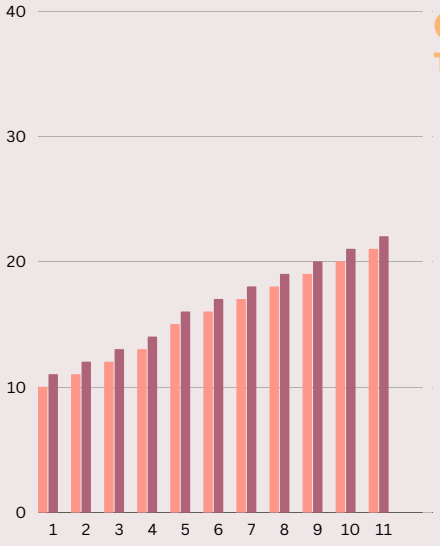


GRÁFICO 3
TESTE_POC.TXT

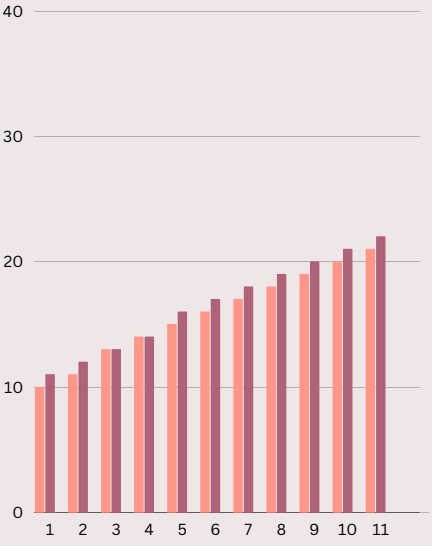


GRÁFICO 4
TESTE_OD.TXT

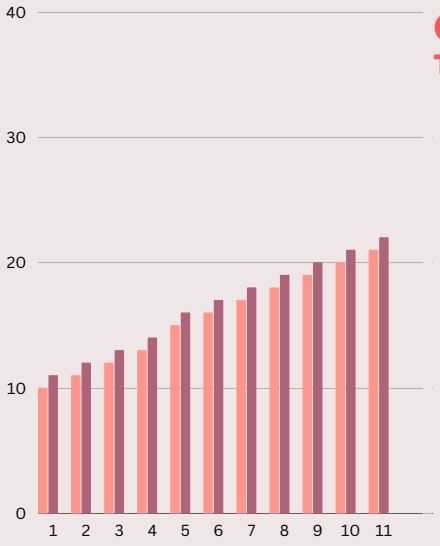
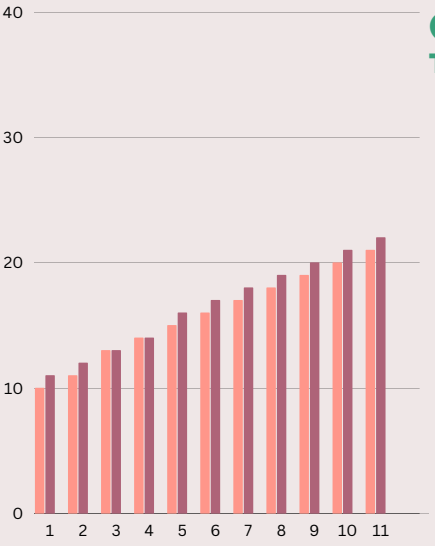
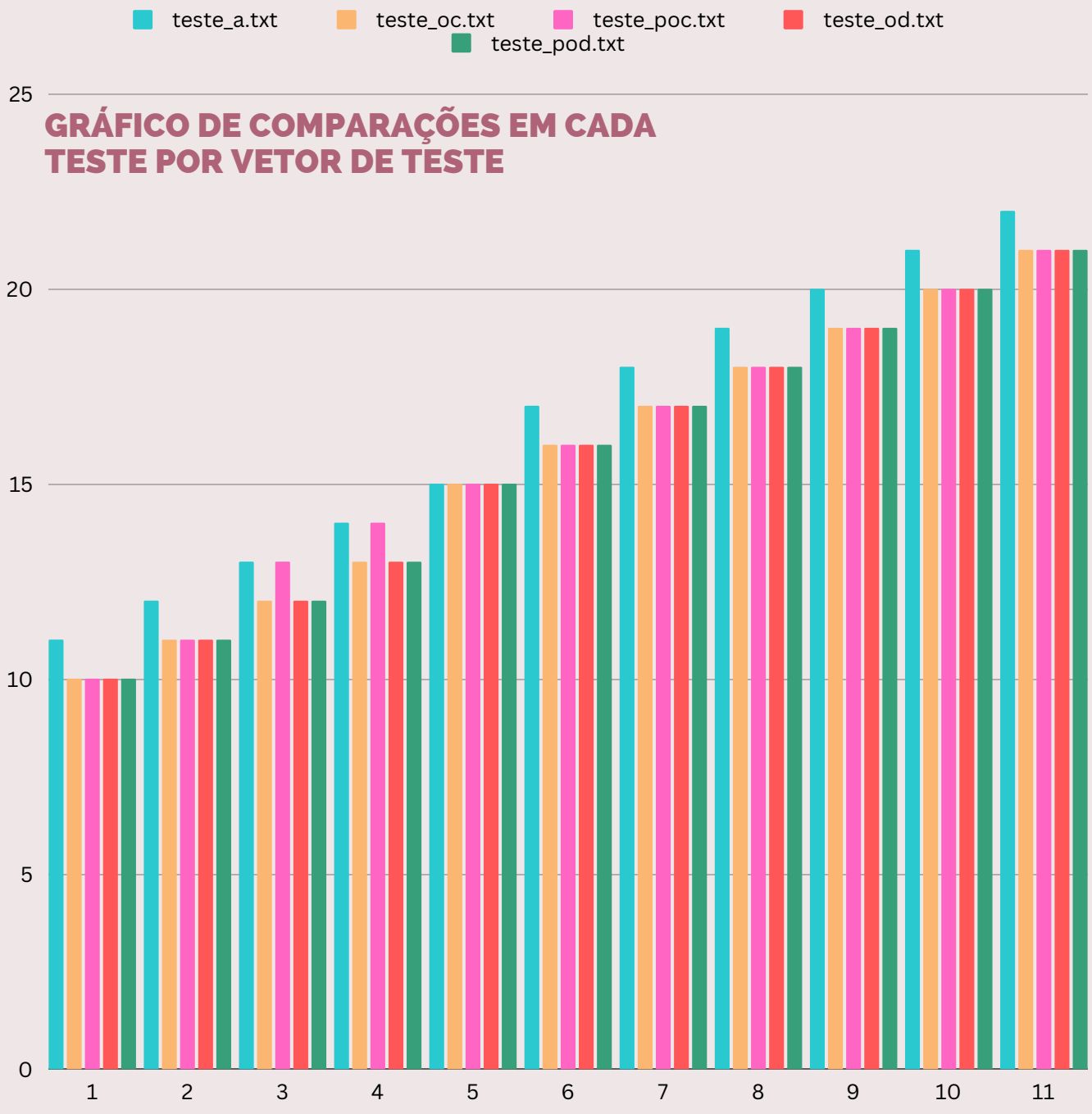


GRÁFICO 5
TESTE_POD.TXT



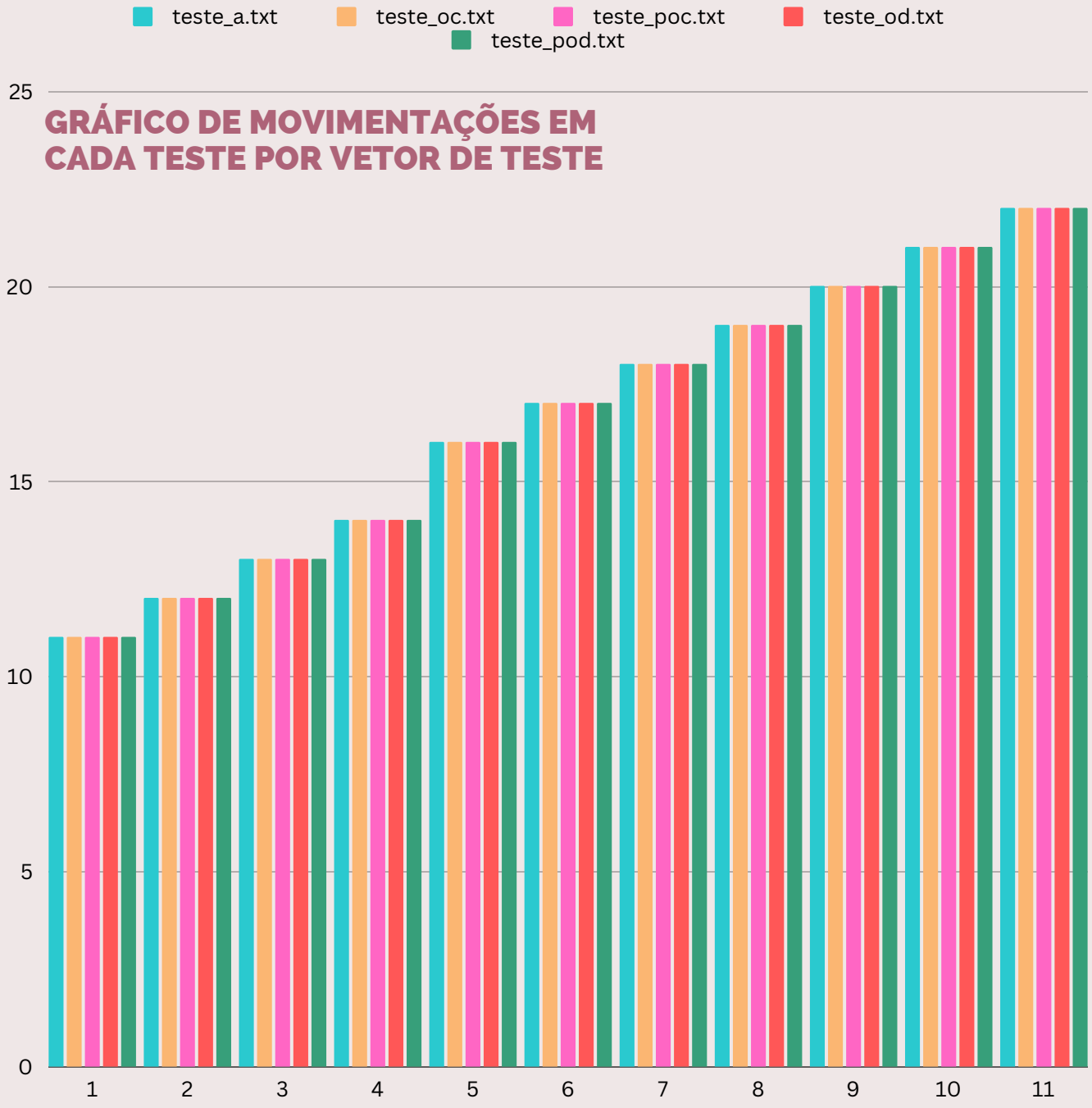
GRÁFICOS



Olhando a quantidade de comparações, observa-se que o Mergesort é mais estável e não varia tanto de acordo com a ordenação do vetor de teste, apesar de ter pontuado melhor com os vetores ordenados de alguma forma. Entretanto, tal diferença não foi tão perceptível no tempo de execução dos testes.

No geral, ao duplicar-se o número de palavras, o número de comparações também foi, aproximadamente, duplicado.

GRÁFICOS

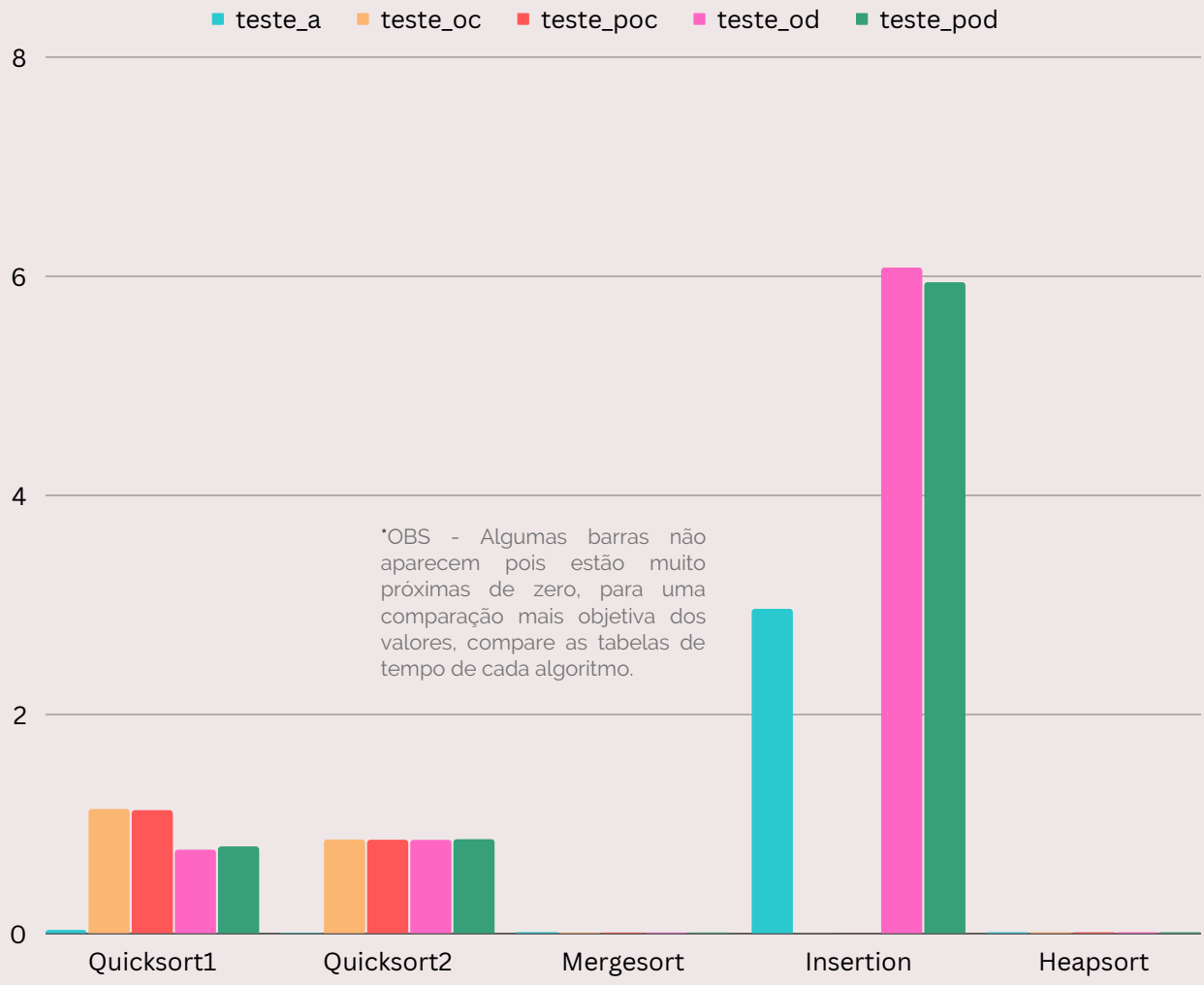


Olhando a quantidade de movimentações, também houve estabilidade, o valor foi aproximadamente duplicado ao duplicar-se a quantidade de palavras testadas.

No teste com o vetor aleatório, o número de movimentações e comparações foi bem próximo, já nos testes com vetores ordenados o número de movimentações foi aproximadamente o dobro do número de comparações.

GRÁFICOS COMPARATIVOS I

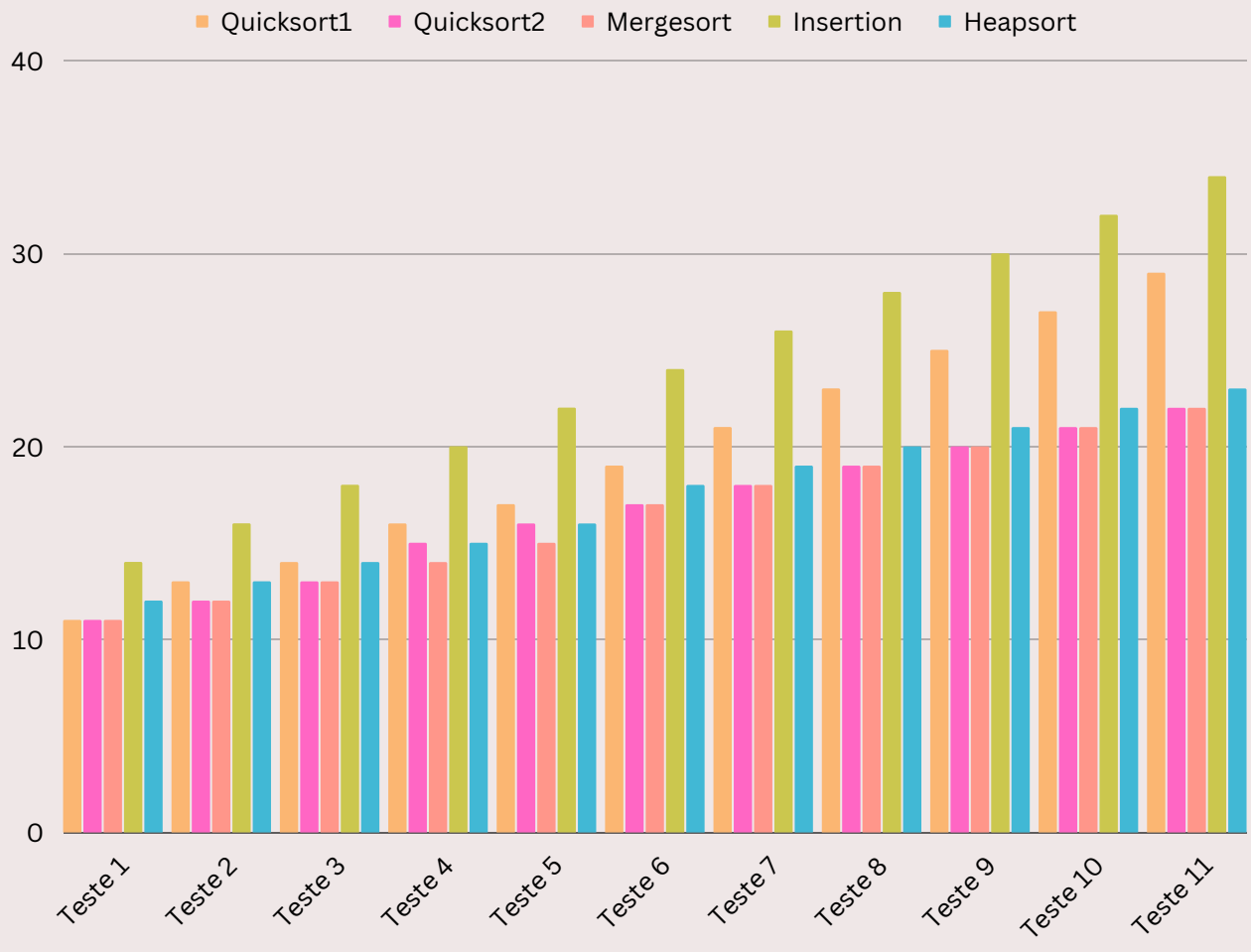
GRÁFICO DE TEMPO DE EXECUÇÃO COMPARANDO OS ALGORITMOS



Como podemos ver no gráfico, os algoritmos com melhor desempenho, independentemente do vetor teste, são o Mergesort e o Heapsort, que são mais estáveis.
Ja os piories desempenhos ficam para o Quicksort 1 nos casos de vetores ordenados e para o Insertion para vetor aleatório e vetores ordenados de forma decrescente, apesar de pontuar excepcionalmente bem para vetores ordenados de forma crescente.

GRÁFICOS COMPARATIVOS II

GRÁFICO DE TOTAL DE COMPARAÇÕES FEITAS COMPARANDO OS ALGORITMOS VETOR: TESTE_A.TXT



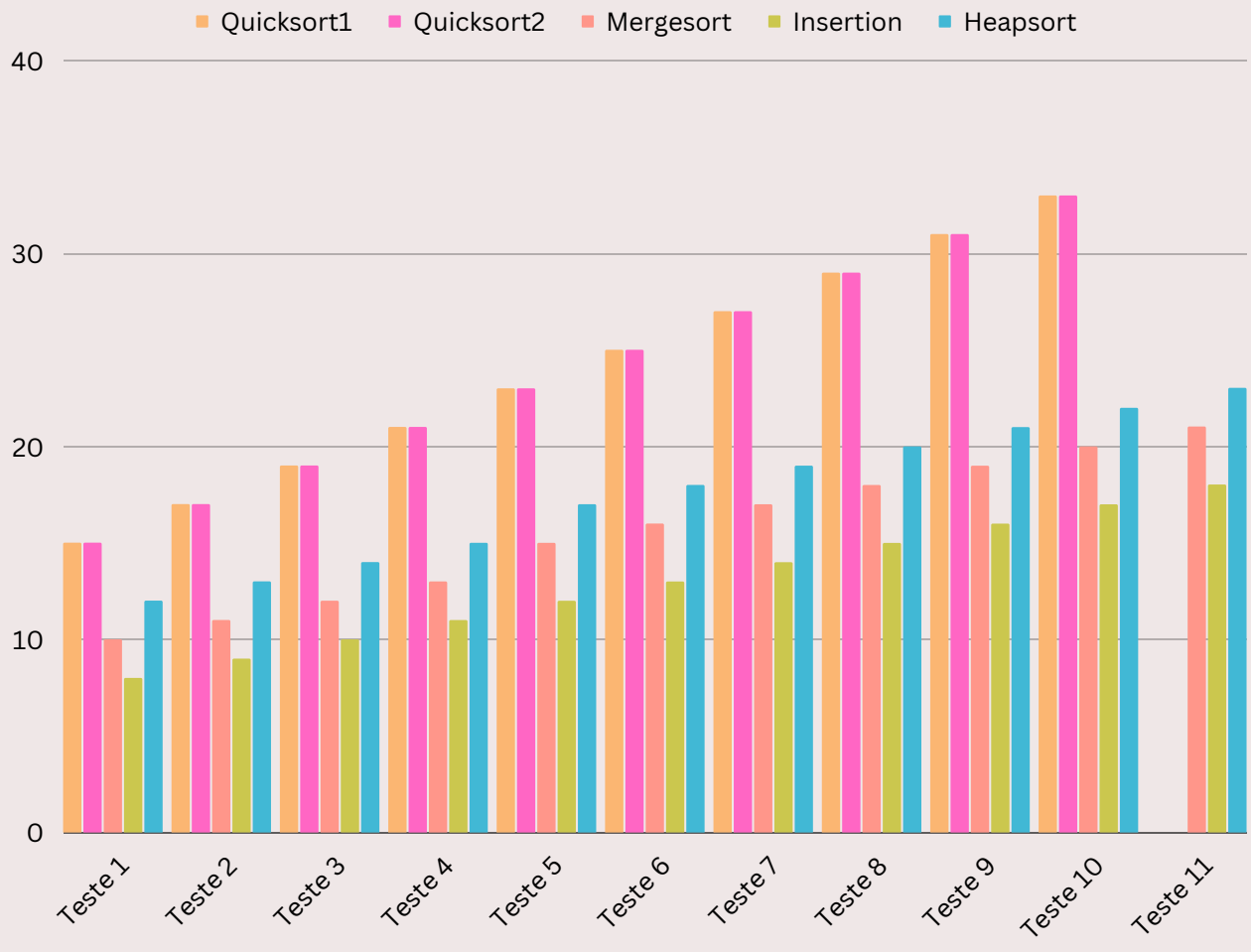
Os gráficos a seguir comparam o desempenho de cada algoritmo nos testes com o vetor indicado na página utilizando o valor de LOG2(N) onde N indica o **número de comparações** expresso na tabela.

Como vemos, os melhores foram o Quicksort 2 e os estáveis (Mergesort e Heapsort). Os piores foram o Quicksort 1 e o Insertion, nos quais o número de comparações quadruplicou a cada teste.

GRÁFICOS COMPARATIVOS II

GRÁFICO DE TOTAL DE COMPARAÇÕES FEITAS COMPARANDO OS ALGORITMOS

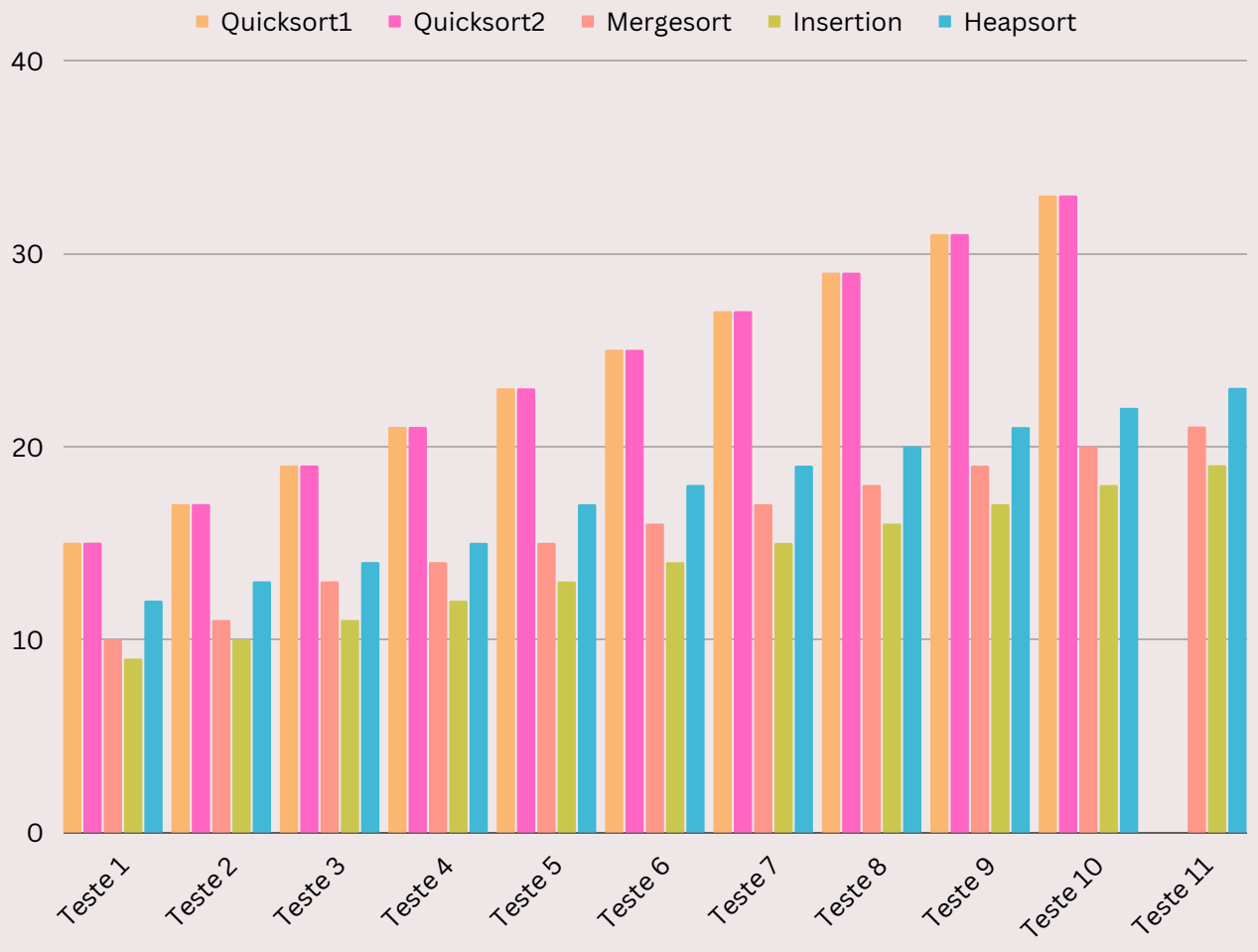
VETOR: TESTE_OC.TXT



Como vemos, o Insertion foi superior a todos os outros e os piores desempenhos ficaram para os Quicksorts. Mergesort e Heapsort mantiveram-se mais estáveis, com o Mergesort ficando em 2º lugar no quesito de desempenho.

GRÁFICOS COMPARATIVOS II

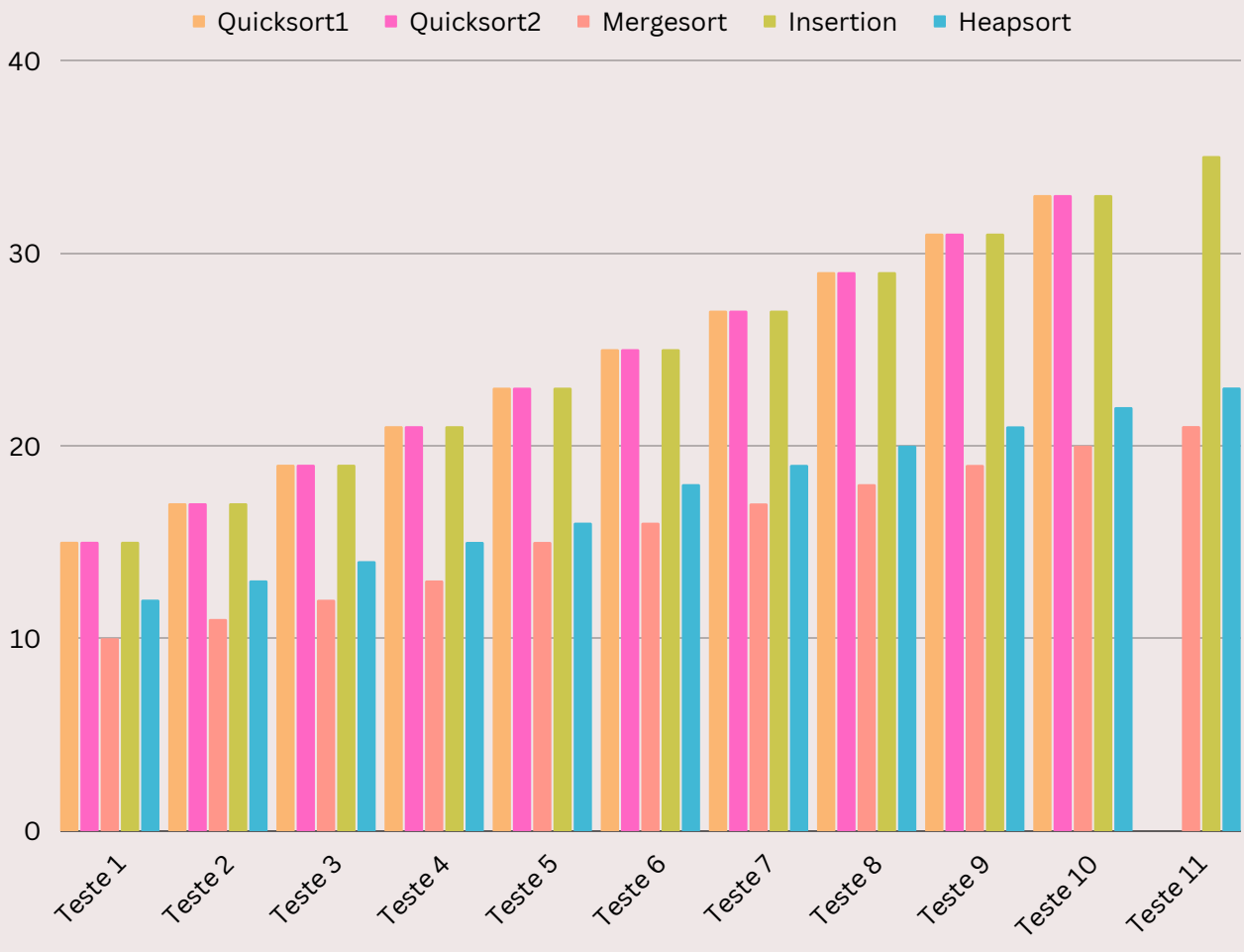
GRÁFICO DE TOTAL DE COMPARAÇÕES FEITAS COMPARANDO OS ALGORITMOS VETOR: TESTE_POC.TXT



Como vemos, o melhor de todos foi o Insertion e o pior foram os Quicksorts. Destaca-se novamente, a estabilidade do Mergesort e do Heapsort.

GRÁFICOS COMPARATIVOS II

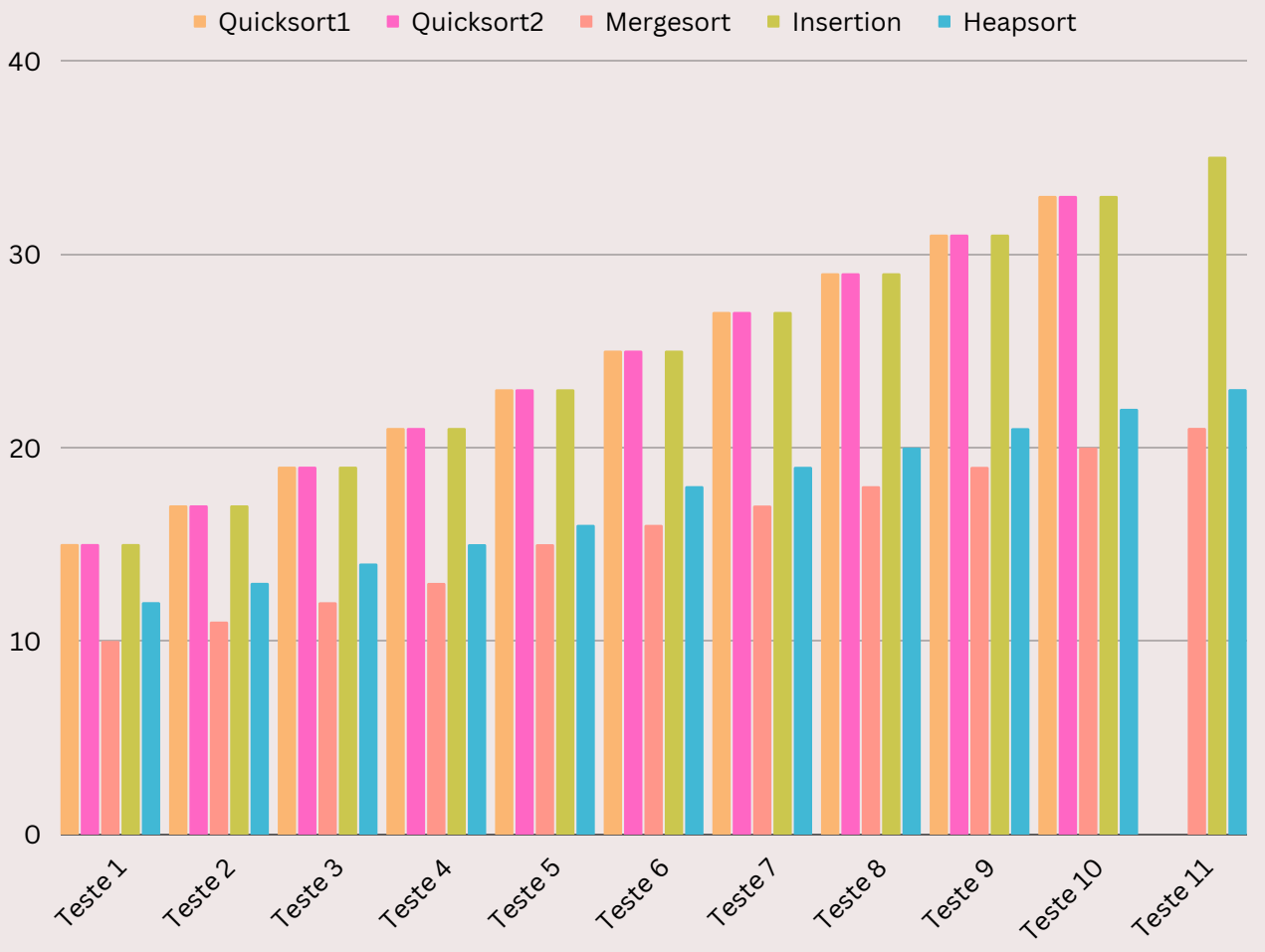
GRÁFICO DE TOTAL DE COMPARAÇÕES FEITAS COMPARANDO OS ALGORITMOS VETOR: TESTE_OD.TXT



Como vemos, o melhor de todos foi o Mergesort e os piores foram os Quicksorts e o Insertion.

GRÁFICOS COMPARATIVOS II

GRÁFICO DE TOTAL DE COMPARAÇÕES FEITAS COMPARANDO OS ALGORITMOS VETOR: TESTE_POD.TXT

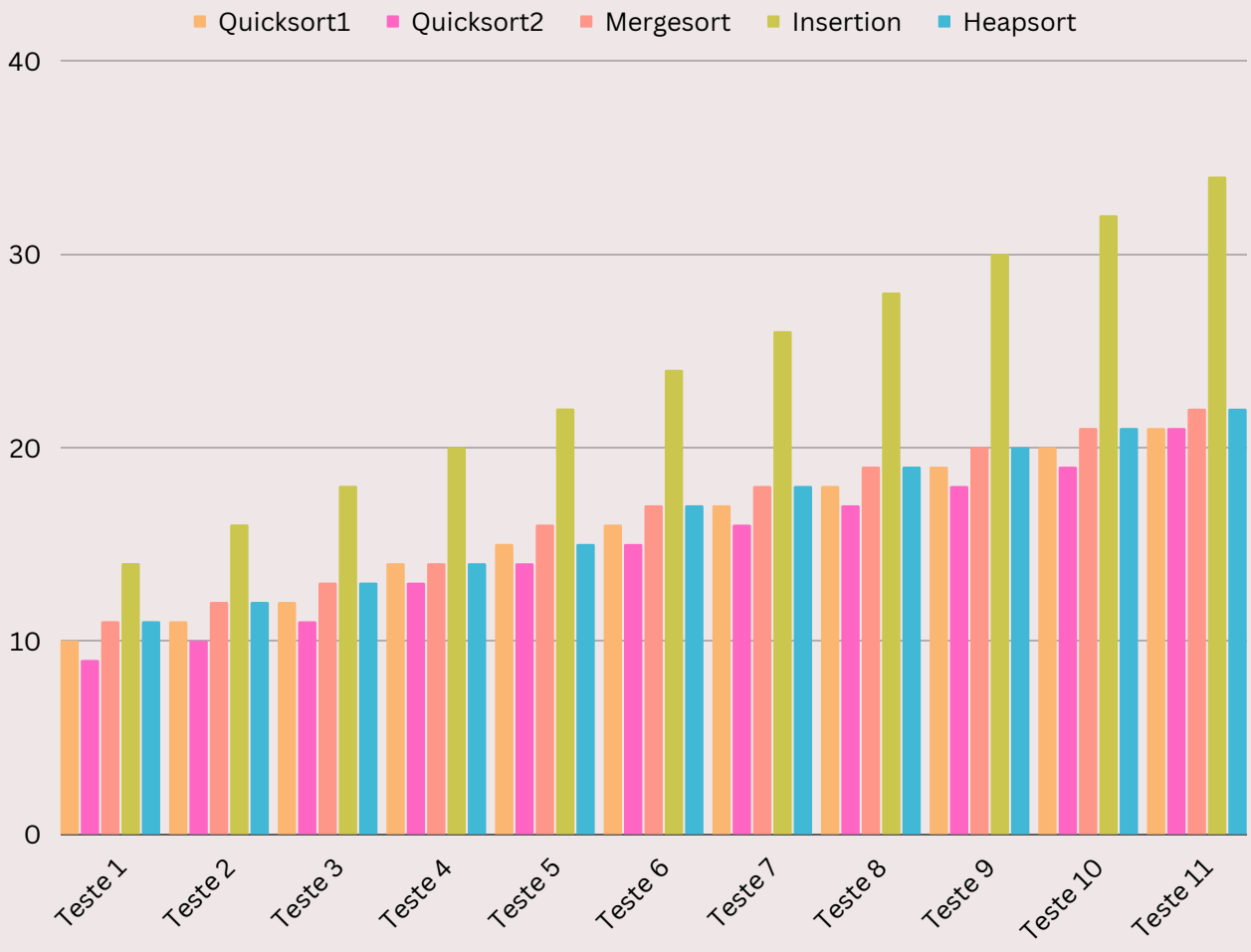


Como vemos, o melhor de todos foi o Mergesort e o piores foram os Quicksorts e o Insertion. O Heapsort, assim como o Mergesort, apresentou um crescimento estável e próximo ao dos outros testes, como se esperava.

GRÁFICOS COMPARATIVOS III

GRÁFICO DE TOTAL DE MOVIMENTAÇÕES FEITAS COMPARANDO OS ALGORITMOS

VETOR: TESTE_A.TXT



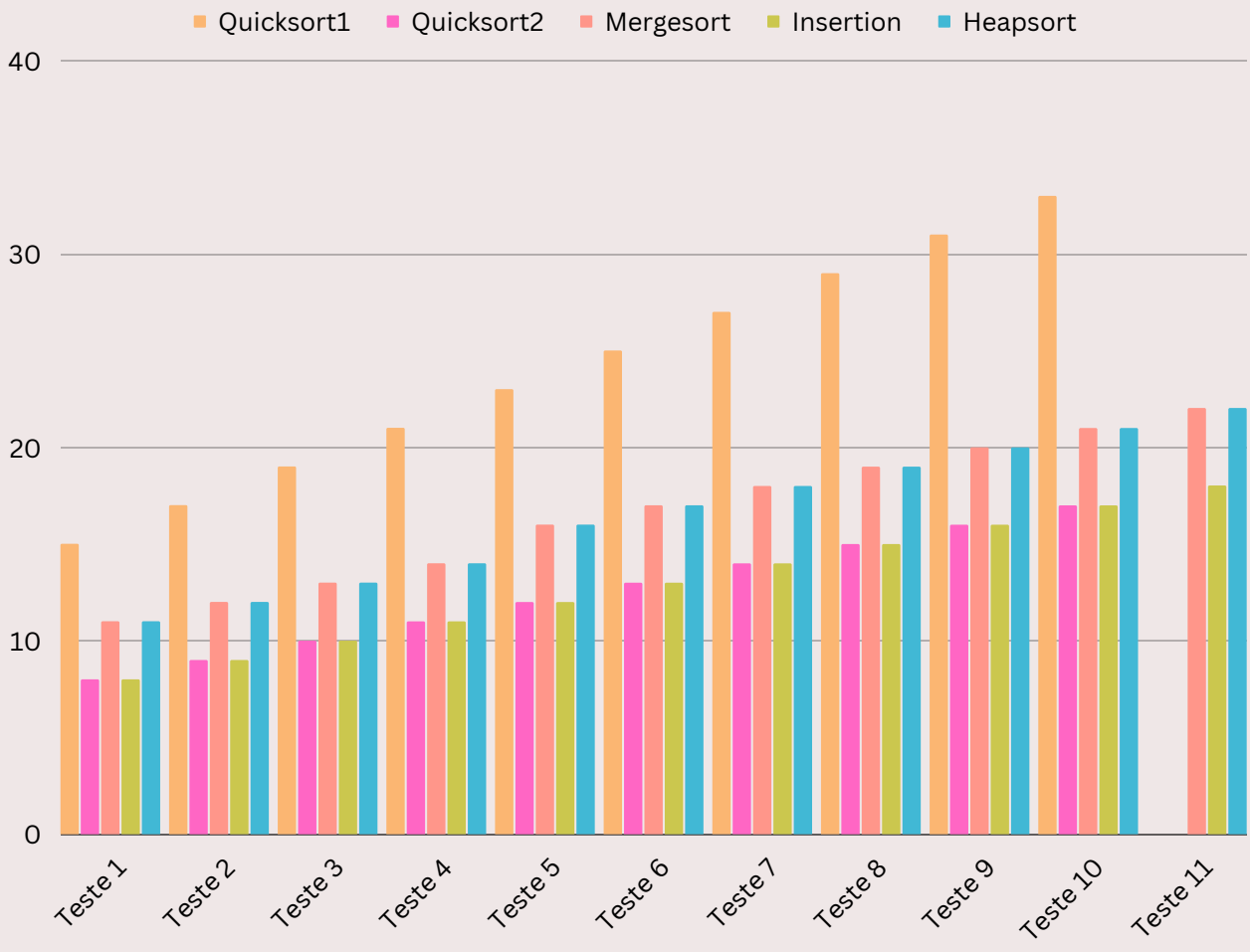
Os gráficos a seguir comparam o desempenho de cada algoritmo nos testes com o vetor indicado na página utilizando o valor de LOG2(N) onde N indica o número de **movimentações / trocas** expresso na tabela.

Como vemos, o melhor de todos foram os Quicksorts e o pior foi o Insertion. Heapsort e Mergesort mantiveram-se estáveis e foram tão bem quanto os Quicksorts.

GRÁFICOS COMPARATIVOS III

GRÁFICO DE TOTAL DE MOVIMENTAÇÕES FEITAS COMPARANDO OS ALGORITMOS

VETOR: TESTE_OC.TXT

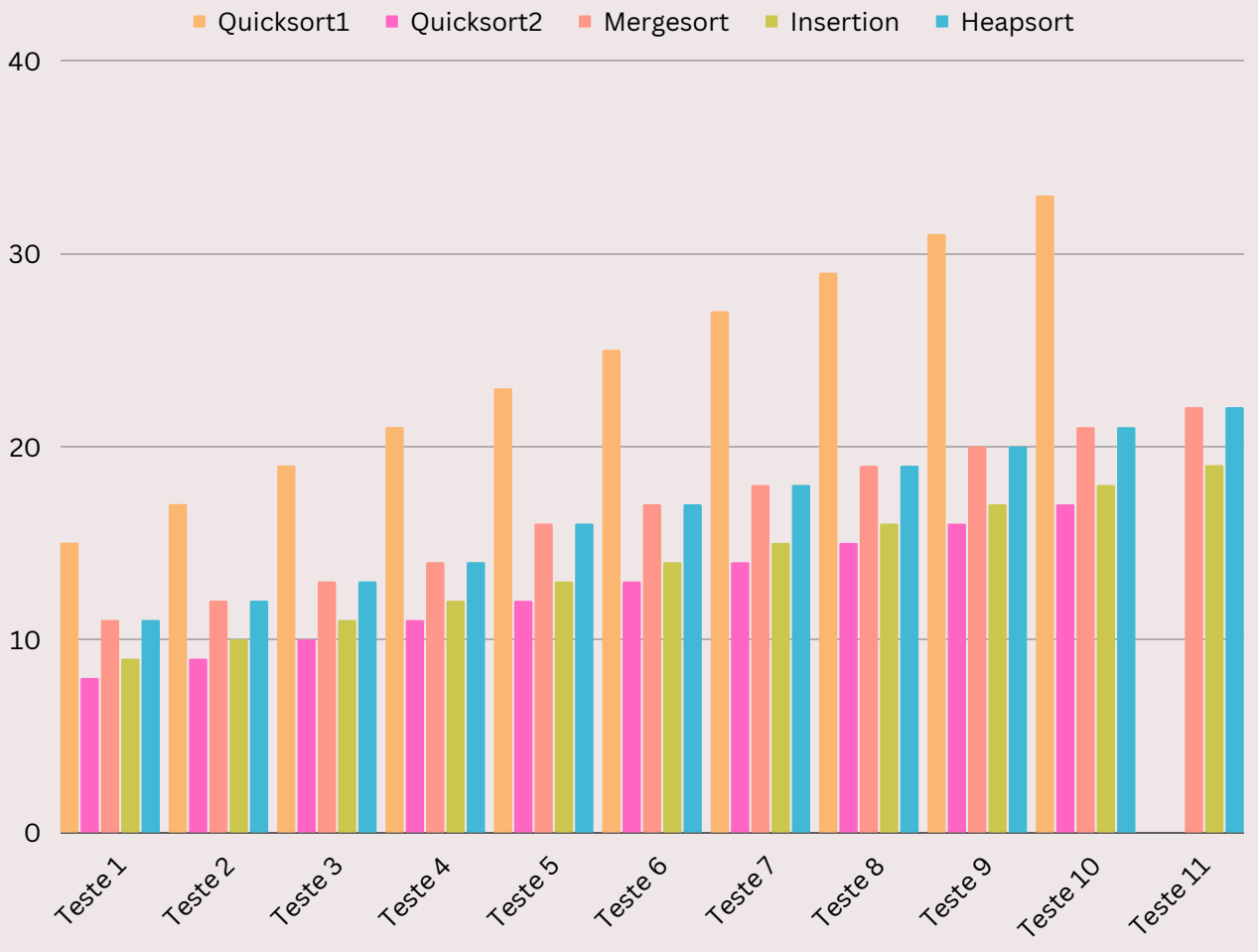


Como vemos, o pior de todos foi o Quicksort 1 e os melhores foram o Insertion e o Quicksort 2 (Porém, vale lembrar que estamos considerando apenas o número de trocas realizadas, já que nesse teste o Quicksort 2 teve um alto número de comparações feitas).

GRÁFICOS COMPARATIVOS III

GRÁFICO DE TOTAL DE MOVIMENTAÇÕES FEITAS COMPARANDO OS ALGORITMOS

VETOR: TESTE_POC.TXT

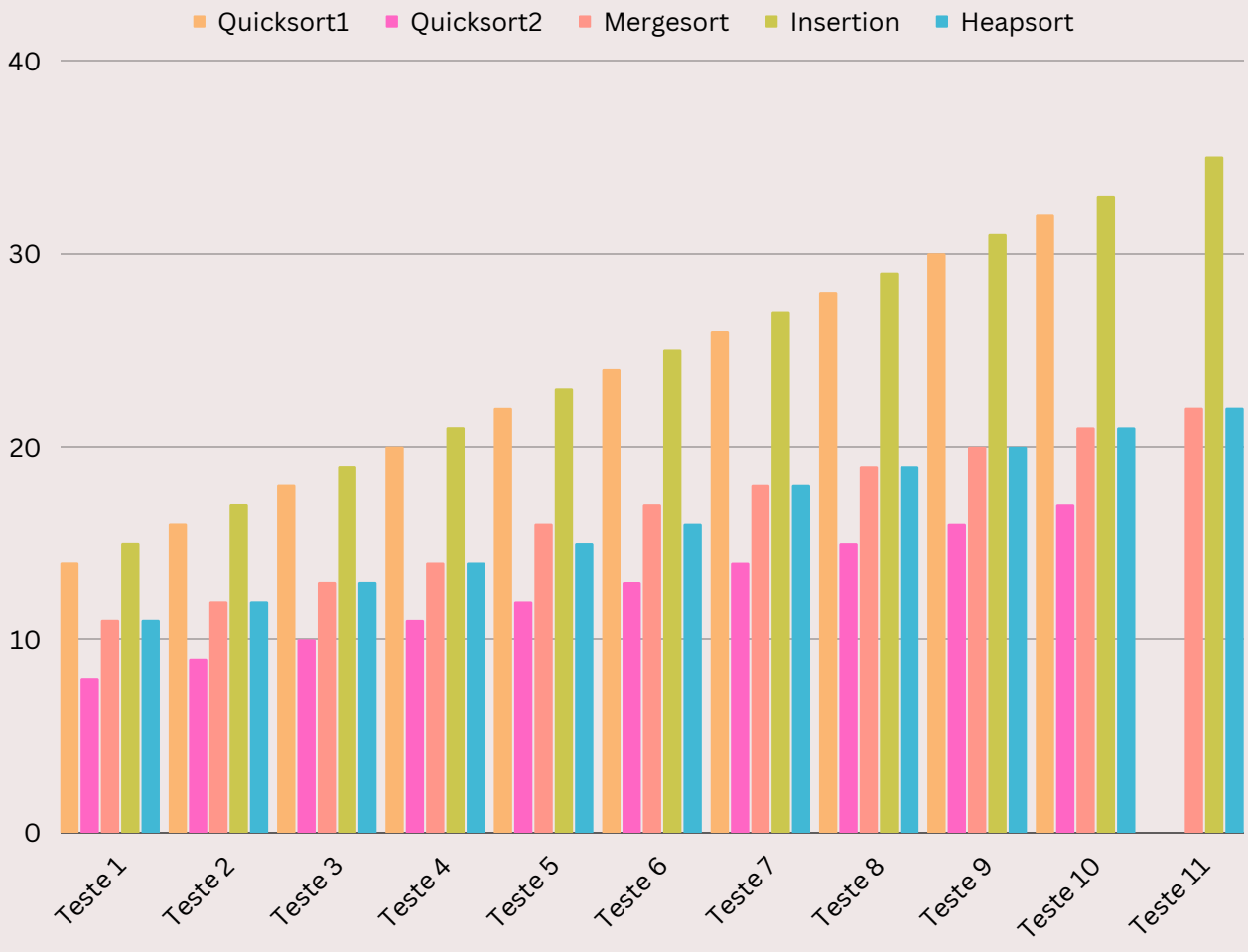


Como vemos, o pior foi o Quicksort 1 e o melhor foi o Quicksort 2, porém, como já foi exposto anteriormente, o número de comparações nesse mesmo teste do Quicksort 2 foi altíssimo, então, destaca-se o desempenho do Insertion, que aliás, diferente dos Quicksorts, rodou todos os testes.

GRÁFICOS COMPARATIVOS III

GRÁFICO DE TOTAL DE MOVIMENTAÇÕES FEITAS COMPARANDO OS ALGORITMOS

VETOR: TESTE_OD.TXT

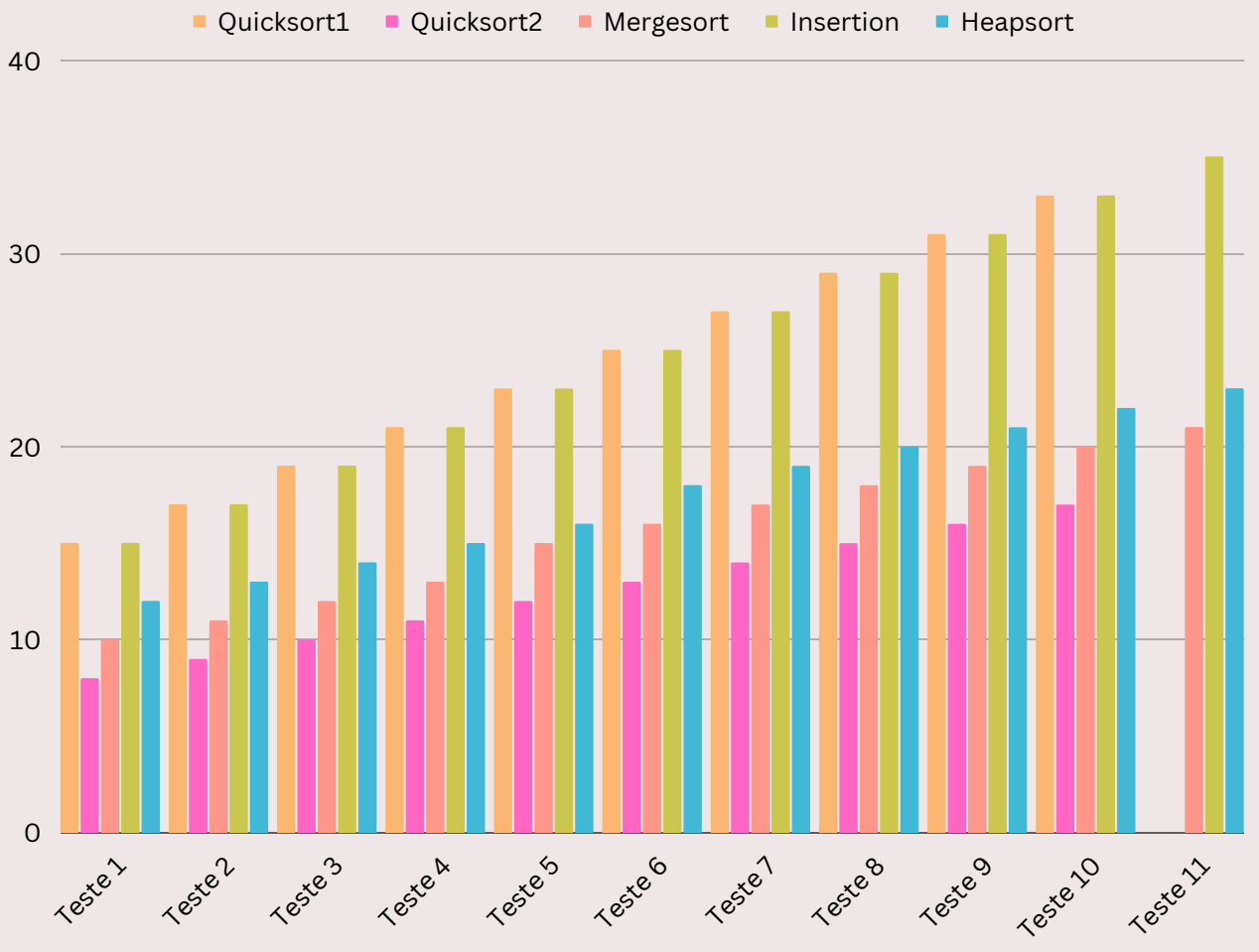


Como vemos, os piores de todos foram o Quicksort 1 e o Insertion. e o melhor foi o Quicksort 2. Porém, tendo em mente que estes resultados contabilizam apenas as trocas realizadas.

GRÁFICOS COMPARATIVOS III

GRÁFICO DE TOTAL DE MOVIMENTAÇÕES FEITAS COMPARANDO OS ALGORITMOS

VETOR: TESTE_POD.TXT



Como vemos, os piores de todos foram o Quicksort 1 e o Insertion. O melhores foram o Mergesort e o Quicksort 2.

CONCLUSÃO

O objetivo da realização de todos os testes era observar o crescimento dos números de comparações e movimentações de cada algoritmo de ordenação e a relação desse número com o número de palavras do vetor de entrada. Para a análise feita a seguir, considere os valores da tabela abaixo que indica o número de comparações e movimentações esperados para alguns algoritmos de ordenação de acordo com o número de elementos a serem ordenados (representado por n).

Algoritmo	Comparações			Movimentações			Espaço	Estável	In situ
	Melhor	Médio	Pior	Melhor	Médio	Pior			
Bubble	$O(n^2)$			$O(n^2)$			$O(1)$	Sim	Sim
Selection	$O(n^2)$			$O(n)$			$O(1)$	Não*	Sim
Insertion	$O(n)$	$O(n^2)$		$O(n)$	$O(n^2)$		$O(1)$	Sim	Sim
Merge	$O(n \log n)$			–			$O(n)$	Sim	Não
Quick	$O(n \log n)$		$O(n^2)$	–			$O(n)$	Não*	Sim
Shell	$O(n^{1.25})$ ou $O(n\ (ln\ n)^2)$			–			$O(1)$	Não	Sim

* Existem versões estáveis.

Fonte: https://www.treinaweb.com.br/blog/conheca-os-principais-algoritmos-de-ordenacao

Além dessas informações, de acordo com o conteúdo do livro **Algoritmos em Linguagem C - Paulo Feofioff**, o Heapsort têm desempenho O(N * LOG2(N)) no geral. Considere também que os dados do Quicksort na tabela acima são mais próximos da implementação do Quicksort 2.

Assim sendo, confira a seguir a conclusão feita dos experimentos realizados.

CONCLUSÃO

QUICKSORTS

Foram testados duas implementações com diferentes versões da função "particiona". O pior desempenho ocorreu com os testes em que o vetor estava ordenado crescentemente, nos quais podemos observar um crescimento do número de comparações / movimentações proporcional a (N^2) como esperado.

No geral, o Quicksort 2 foi o de melhor de desempenho entre os dois e para o caso aleatório o crescimento foi aproximadamente proporcional a (N * LOG2(N)). Conclusão, é um algoritmo de ordenação rápido, mas para os casos em que o vetor está ordenado ou parcialmente ordenado seu desempenho cai um pouco.

HEAPSORT

Comportou-se de maneira mais estável pois as características de ordenação do vetor não influenciaram muito o desempenho do algoritmo, sendo que o número de comparações / movimentações foi proporcional ao esperado, (N * LOG2(N)).

Ou seja, ele é confiável para casos em que não se saiba muito sobre o vetor de entrada, pois há uma garantia de estabilidade em seus resultados.

CONCLUSÃO

INSERTION

Teve, exceto no caso do vetor de entrada já ordenado crescentemente, o pior desempenho no geral com maior número de movimentações e comparações. O número de comparações e movimentações foi proporcional a (N^2) nos vetores ordenados de forma decrescente. No caso aleatório, os valores variaram mais, entretanto ficaram mais próximos do pior caso.

No caso em que o vetor estava ordenado de forma crescente (completamente ou parcialmente), o número de comparações e movimentações foi proporcional a N. Ou seja, caso o vetor esteja ordenado, dependendo da implementação do insertion (Considere um insertion que ordene de forma decrescente por exemplo), a melhor escolha é o Insertion.

MERGESORT

Assim como Heapsort, não variou muito com as alterações de características de ordenação dos vetores de teste, assim compartilha muitas semelhanças com o Heapsort em questão de desempenho. O número de comparações e movimentações, no geral, foram proporcionais a (N * LOG2(N)) como esperado.

Ou seja, também é um algoritmo confiável para casos em que não se saiba muito sobre os vetores de entrada, porém, a implementação feita depende do auxílio de vetores auxiliarem, o que ocupa mais memória.