THIAGO MORAES

MATRÍCULA 21452625

TP14-LPAV

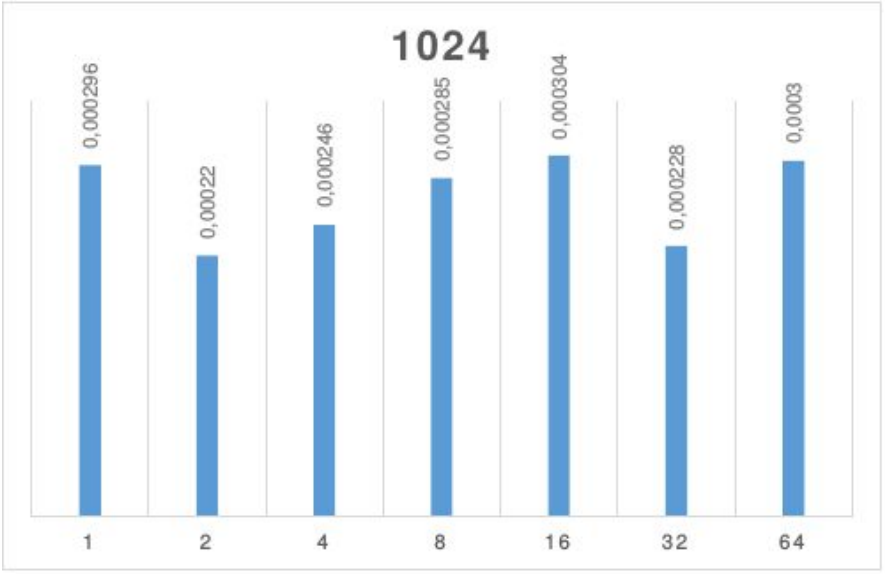
PTHREADS

1) QUESTÃO

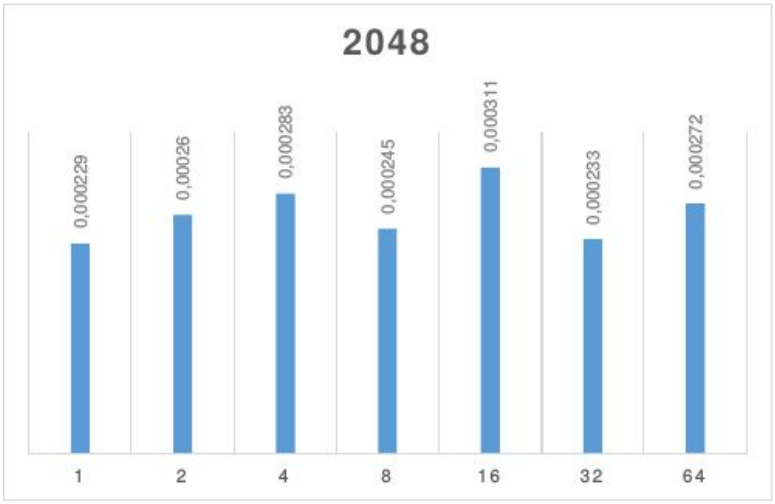
N/K	1024	2048	4096	8192	16384	32768	65536
1	0,000296	0,000229	0,000282	0,000218	0,000303	0,000194	0,000199
2	0,00022	0,00026	0,000332	0,000235	0,000222	0,000208	0,000295
4	0,000246	0,000283	0,000235	0,000282	0,00021	0,000261	0,000142
8	0,000285	0,000245	0,000182	0,000164	0,000079	0,000236	0,000169
16	0,000304	0,000311	0,00033	0,000234	0,000305	0,00026	0,000297
32	0,000228	0,000233	0,000278	0,000306	0,000245	0,000283	0,000298
64	0,0003	0,000272	0,000297	0,000092	0,000294	0,000296	0,000237

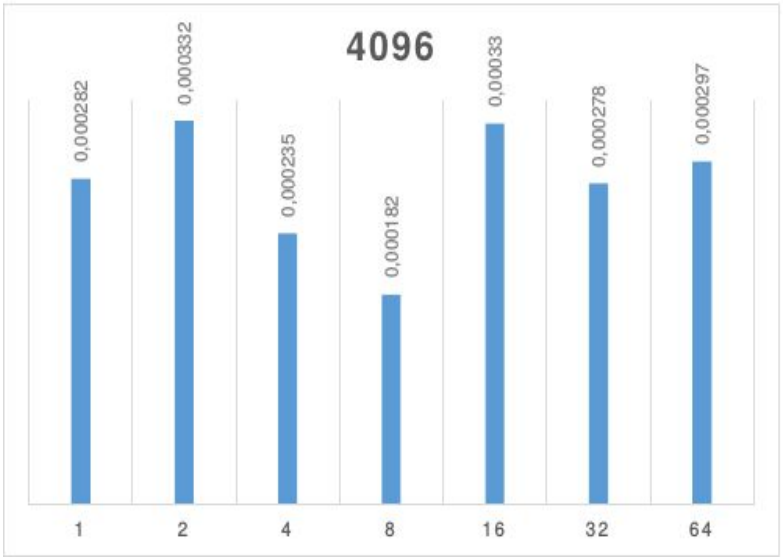
Gráficos

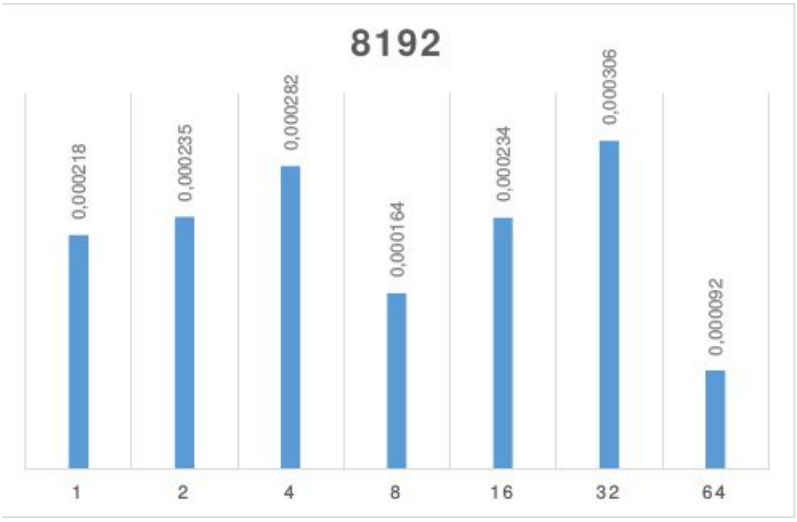
Obs: Os gráficos em linhas ficaram muito ruim para visualização, mesmo em escala logaritma o entendimento ficou ruim, então por isso plotei os gráficos nessa forma abaixo, os valores de K estão na parte de cima do gráfico, e em cada barra azul contém sua thread e seu tempo calculado, exemplo 1 - 0,000296.

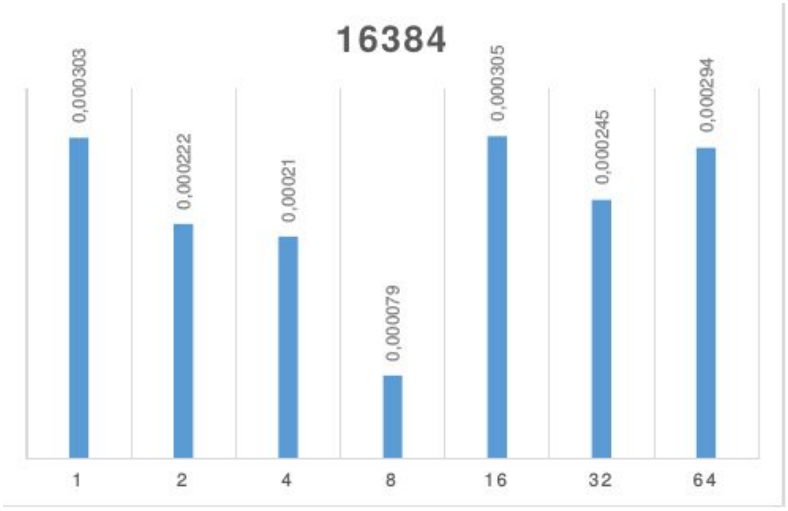


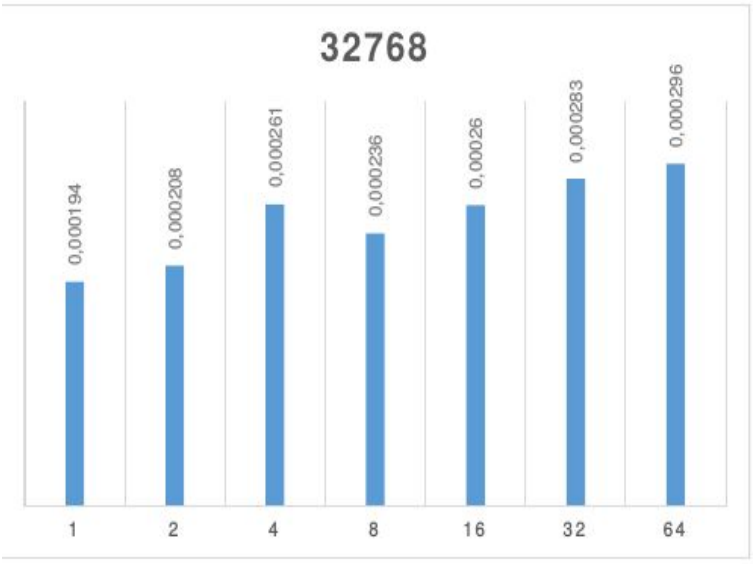
K = 1024 teve meu melhor desempenho com 2 threads

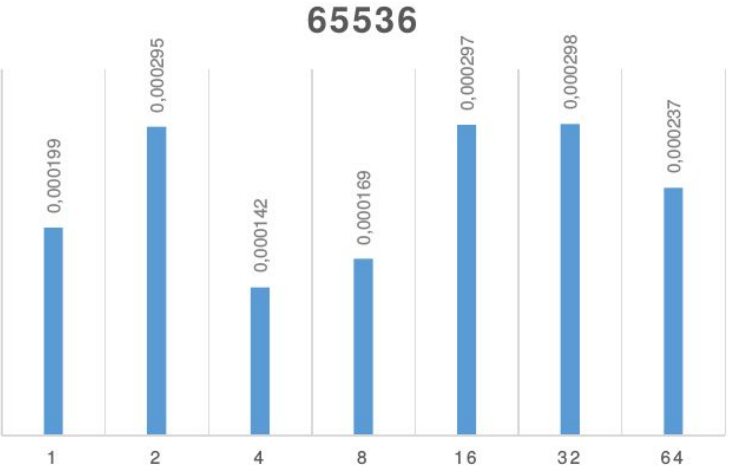

2048
0,000229
0,00026
0,000283
0,000245
0,000311
0,000233
0,000272
1
2
4
8
16
32
64


4096
0,000282
0,000332
0,000235
0,000182
0,00033
0,000278
0,000297
1
2
4
8
16
32
64


8192
0,000218
0,000235
0,000282
0,000164
0,000234
0,000306
0,000092
1
2
4
8
16
32
64


16384
0,000303
0,000222
0,00021
0,000079
0,000305
0,000245
0,000294
1
2
4
8
16
32
64


32768
0,000194
0,000208
0,000261
0,000236
0,00026
0,000283
0,000296
1
2
4
8
16
32
64

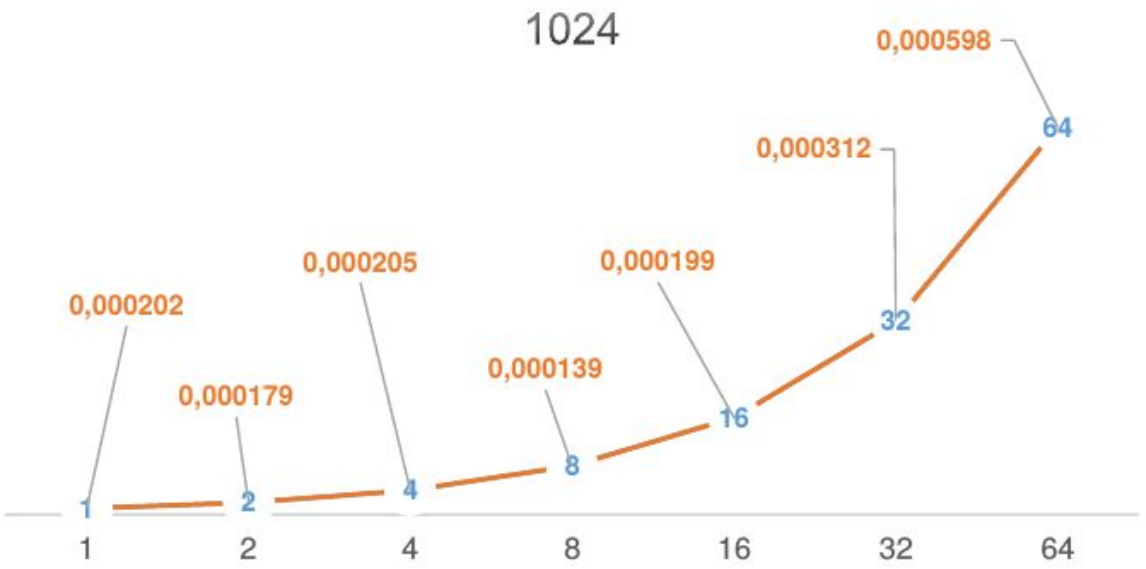

65536
0,000199
0,000295
0,000142
0,000169
0,000297
0,000298
0,000237
1
2
4
8
16
32
64

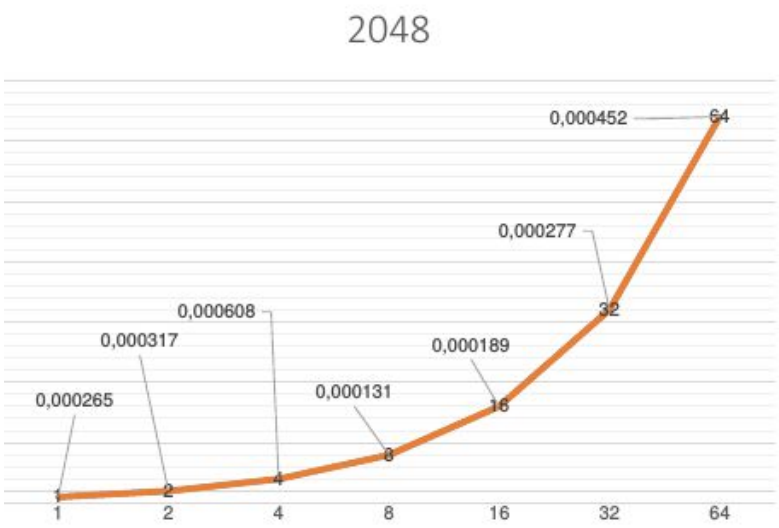
Questão 2)

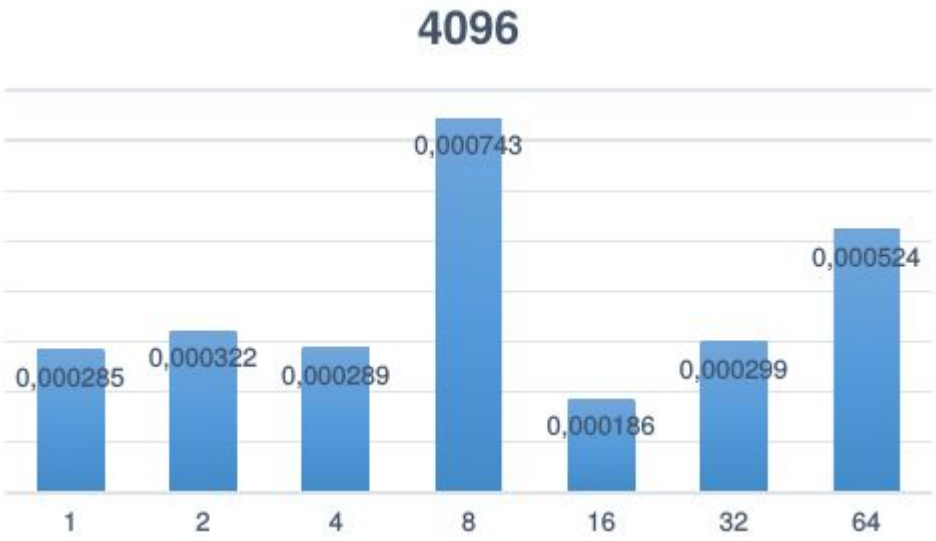
Tabela

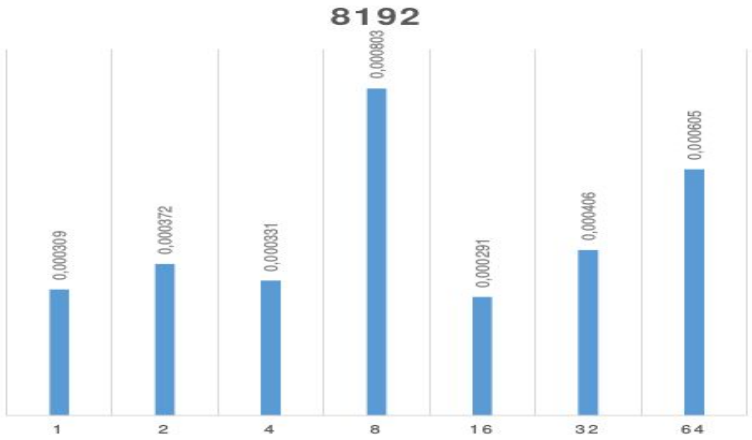
N/K	1024	2048	4096	8192	16384	32768	65536
1	0,000202	0,000265	0,000285	0,000309	0,000591	0,000553	0,001463
2	0,000179	0,000317	0,000322	0,000372	0,000603	0,000267	0,001802
4	0,000205	0,000608	0,000289	0,000331	0,000446	0,000682	0,001058
8	0,000139	0,000131	0,000743	0,000803	0,000532	0,0001182	0,001336
16	0,000199	0,000189	0,000186	0,000291	0,000746	0,00022	0,001486
32	0,000312	0,000277	0,000299	0,000406	0,000314	0,00065	0,000839
64	0,000598	0,000452	0,000524	0,000605	0,000494	0,000544	0,000686

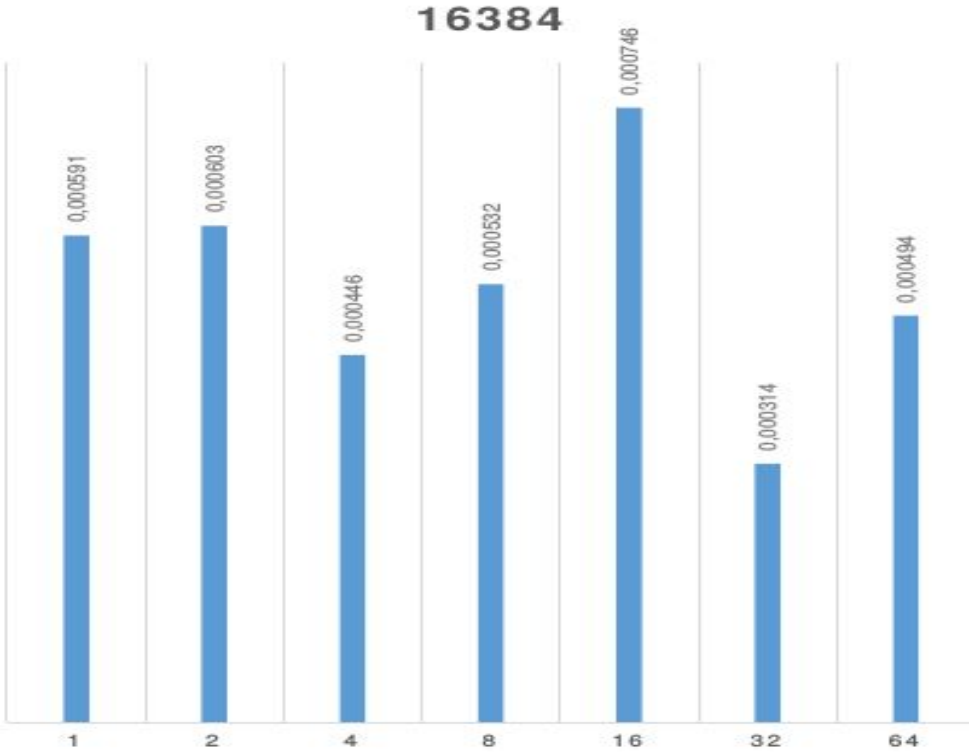
Gráficos



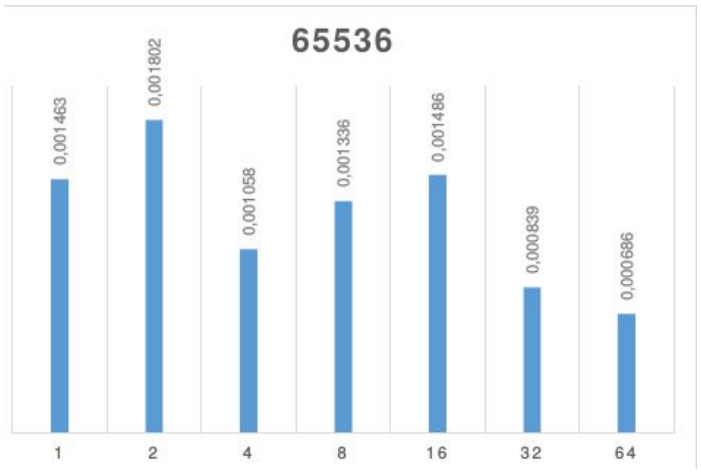

2048
0,000452
64
0,000277
32
0,000608
0,000317
0,000189
0,000131
0,000265
16
8
4
2
1
1
2
4
8
16
32
64


4096
0,000743
0,000524
0,000322
0,000285
0,000289
0,000299
0,000186
1
2
4
8
16
32
64


8192
0,000309
0,000372
0,000331
0,000803
0,000291
0,000406
0,000605
1
2
4
8
16
32
64


16384
0,000591
0,000603
0,000446
0,000532
0,000746
0,000314
0,000494
1
2
4
8
16
32
64


32768
0,000553
0,000267
0,000682
0,0001182
0,00022
0,00065
0,000544
1
2
4
8
16
32
64

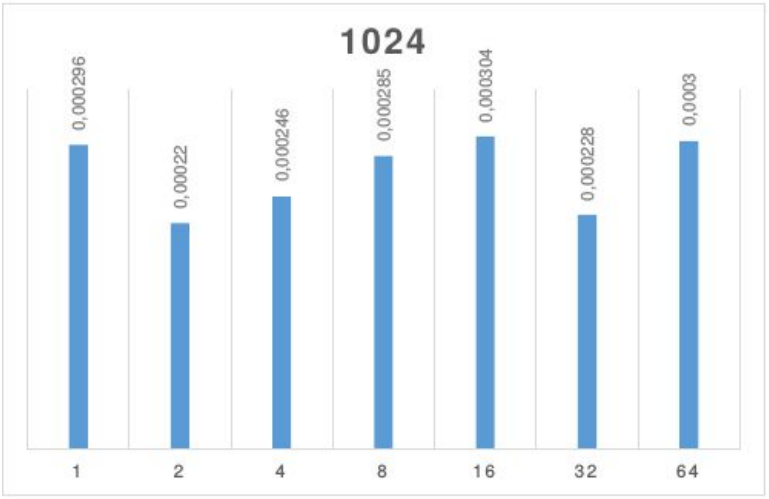

65536
0,001463
0,001802
0,001058
0,001336
0,001486
0,000839
0,000686
1
2
4
8
16
32
64

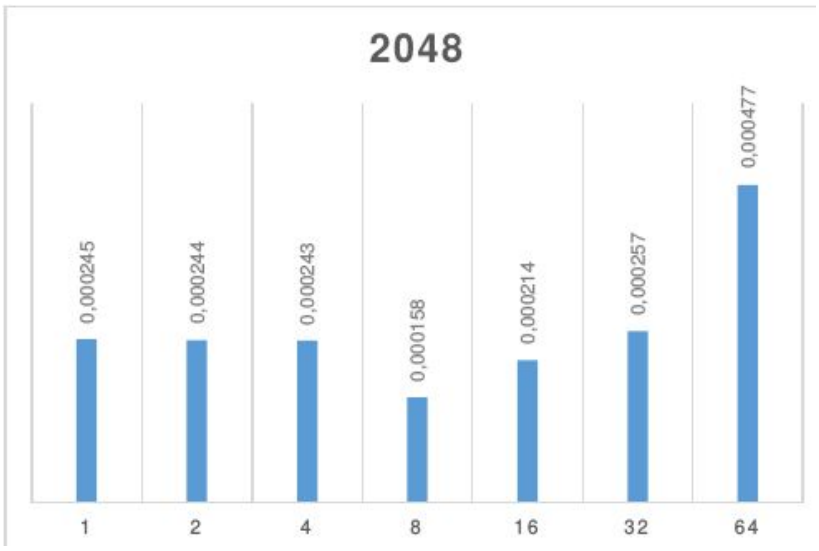
Questão 3)

Tabela

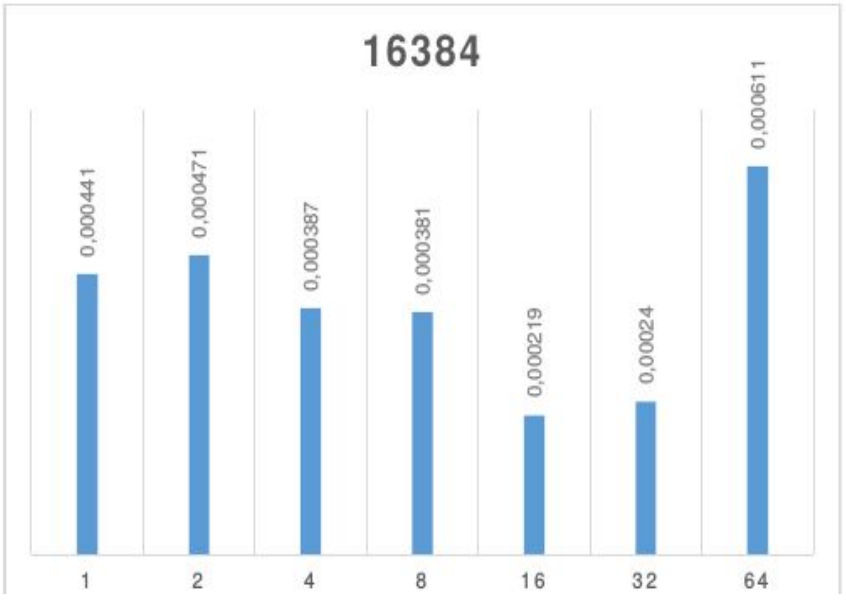
N/K	1024	2048	4096	8192	16384	32768	65536
1	0,000242	0,000245	0,000319	0,000244	0,000441	0,000308	0,000976
2	0,000203	0,000244	0,000128	0,000245	0,000471	0,000581	0,000552
4	0,000235	0,000243	0,00033	0,000316	0,000387	0,000614	0,001033
8	0,000062	0,000158	0,000355	0,00072	0,000381	0,000406	0,000915
16	0,000167	0,000214	0,000329	0,00025	0,000219	0,000232	0,000099
32	0,000369	0,000257	0,00026	0,000286	0,00024	0,000264	0,000325
64	0,000729	0,000477	0,000664	0,000476	0,000611	0,0005	0,000625

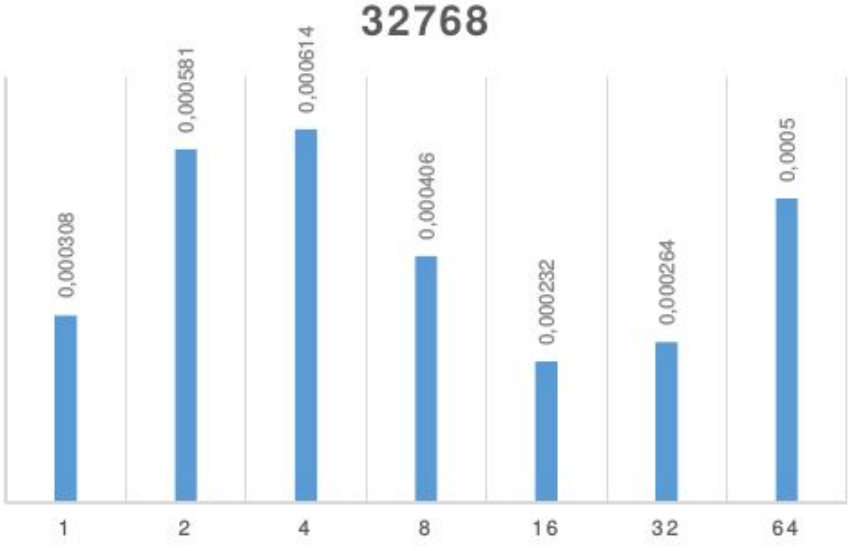
Gráficos






4096
0,000319
0,000128
0,00033
0,000355
0,000329
0,00026
0,000664
1
2
4
8
16
32
64


16384
0,000441
0,000471
0,000387
0,000381
0,000219
0,00024
0,000611
1
2
4
8
16
32
64

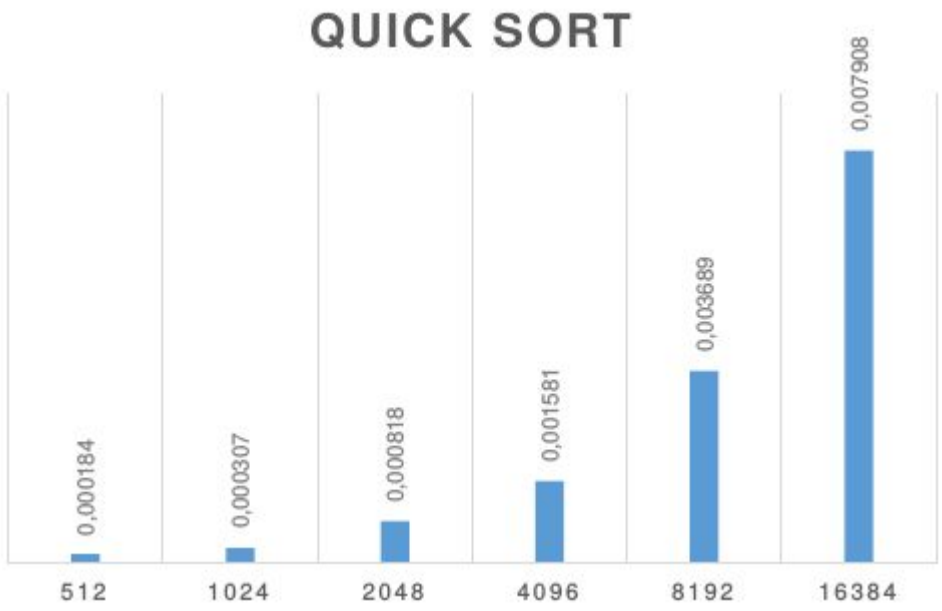

32768
0,000308
0,000581
0,000614
0,000406
0,000232
0,000264
0,0005
1
2
4
8
16
32
64

Em relação a questão 1 o que posso relatar depois de analisar foi que obtive um bom resultado mesmo com um vetor grande, se observamos um k = 1024 tivemos o tempo 0,000296 com uma thread e para k = 65536 com uma thread foi 0,000199, ou seja um tempo menor, concluir que o quanto maior tamanho do vetor mais rápido foi as operações.

Na questão 2, foi o contrário da questão 1, tivemos tempo menores com vetores menores, se considerarmos o bom senso isso parece lógico, exemplo k = 1024 uma thread tivemos o tempo de 0,000202 e com uma thread para um vetor grande de 65536 tivemos um aumento considerável de 00,001463.

Já na questão 3, não consegui encontrar um padrão exato de variação, algumas vezes com um vetor pequeno (1024) tivemos tempo pequenos para uma thread e mesmo aumentando o número de thread para esse vetor o tempo não diminuía, concluir que processos com muitas threads ficam mais lentos.

Questão 4)



Sobre o QuickSort, seu desempenho com qualquer um dos tamanhos teve bons resultados, acredito que mesmo com os valores do vetor estando aleatórios, ele executou em tempo consideravelmente bom.