

Aluno: Jhonatan Guilherme de Oliveira Cunha **RA:** 2135590

Disciplina: Algoritmo e Estrutura de Dados 2

5)

TEMPO DE EXECUÇÃO PARA K = 1

	N = 1000	N = 10000	N = 100000	N = 500000	N = 1000000
SelectionMinK	0	0,0010	0,0010	0,0150	0,0090
QuickMinK	0	0,0000	0,0000	0,0000	0,0040
HeapMinK	0	0,0010	0,0050	0,0160	0,0220

TEMPO DE EXECUÇÃO PARA K = (N/3)

	N = 1000	N = 10000	N = 100000	N = 500000	N = 1000000
SelectionMinK	0,0030	0,1300	14,6620	325,2420	1325,7930
QuickMinK	0,0000	0,000	00,0160	0,0160	0,0780
HeapMinK	0,0120	0,3030	31,5690	751,4590	3050,3130

TEMPO DE EXECUÇÃO PARA K = (N/2)

	N = 1000	N = 10000	N = 100000	N = 500000	N = 1000000
SelectionMinK	0,0020	0,2290	17,7360	440,8150	
QuickMinK	0,0000	0,0000	00,0000	00,0310	
HeapMinK	0,0060	0,4530	40,1100	1015,8840	

TEMPO DE EXECUÇÃO PARA $K = N$

	N = 1000	N = 10000	N = 100000	N = 500000	N = 1000000
SelectionMinK	0,0000	0,2960	23,3600	595,7760	
QuickMinK	0,0000	0,0000	0,0000	0,0310	
HeapMinK	0,0000	0,5470	53,9460	1370,1990	

6) Você notou algum padrão nos resultados obtidos no item 5? Explique o que você descobriu

Percebi que o quickSortMinK em vetores desordenado executa muito mais rápido que os outros dois algoritmos mesmo k sendo diferente de 1.

O padrão que eu percebi analisando o tempo de execução desses algoritmos foi que o heapMinK na maioria dos casos levou quase ou mais que o dobro de tempo que o algoritmo de selectionMinK.