Universidade Tecnológica Federal do Paraná
Disciplina: Algoritmos e Estruturas de Dados 2
Aluno: Jean Carlos Martins Miguel RA 1640593
Professor: Dr. Juliano Henrique Foleis

Semana 5: Busca em Vetor – Busca Sequencial e Busca Binária

5. Execute os três algoritmos sugeridos acima para buscar a k-ésima menor chave de um vetor aleatório gerado com a função int* random_vector_unique_elems(int n, int seed) com n = 1000, 10000, 100000, 500000 e seed = 42. Preencha as tabelas a seguir com o tempo de execução dos seguintes casos de testes (só precisa executar o SelectionMinK até p/ k=10000)

K = 1

	1000	10000	100000	500000	1000000
SelectionMink	0.120000	0.365000	1.500000		
HeapMink	0.090000	1.555000	12.199000	14.954000	29.326000
QuickMinK	0.107000	0.288000	1.267000	1.533000	3.063000

k = n/3

	1000	10000	100000	500000	1000000
SelectionMink	1.335000	128.955000	10804.047000		
HeapMink	10.947000	170.726000	14298.790000	372074.426000	
QuickMinK	0.273000	1.640000	11.924000	11.788000	38.995000

k = n /2

	1000	10000	100000	500000	1000000
SelectionMink	9.750000	245.300000	18720.656000		
HeapMink	11.237000	278.552000	22875.182000	557673.394000	
QuickMinK	0.288000	1.650000	14.282000	14.836000	32.819000

k = n

	1000	10000	100000	500000	1000000
SelectionMink	18.024000	416.409000	38725.488000		
HeapMink	16.93600	276.699000	25495.103000		
QuickMinK	0.186000	0.760000	6.291000	13.813000	16.849000

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

R: O quickSort em vetores desordenados executa mais rapidamente que o selectionMink e o HeapMink, conforme aumenta-se o k no algortimo HeapMink demora mais tempo para ser executado, até duas vezes mais em alguns casos em comparação com os outros dois.