Atividade 5 – Algoritmo de busca
Algoritmo e estrutura de dados 2 – UTFPR-CM – 3° período
Nome: Matheus Santos de Andrade RA: 2304570

4.

TABELA 1 – k = 1

	N = 1000	N = 10000	N = 100000	N = 500000	N = 1000000
SelectionMinK	**0.000000**	**0.000000**	**0.000000**	**0.000000**	**0.000000**
QuickMinK	**0.000000**	**0.000000**	**0.008000**	**0.008000**	**0.013000**

TABELA 2 – k = n/3

	N = 1000	N = 10000	N = 100000	N = 500000	N = 1000000
SelectionMinK	**0.000000**	**0.056000**	**//**	**//**	**//**
QuickMinK	**0.000000**	**0.000000**	**0.004000**	**0.027000**	**0.064000**

TABELA 3 – k = n/2

	N = 1000	N = 10000	N = 100000	N = 500000	N = 1000000
SelectionMinK	**0.000000**	**0.076000**	**//**	**//**	**//**
QuickMinK	**0.000000**	**0.000000**	**0.016000**	**0.041000**	**0.088000**

TABELA 4 – k = n

	N = 1000	N = 10000	N = 100000	N = 500000	N = 1000000
SelectionMinK	**0.000000**	**0.101000**	**//**	**//**	**//**
QuickMinK	**0.000000**	**0.000000**	**0.000000**	**0.001000**	**0.002000**

5.

O método *SelectionMinK* possui um tempo de execução ridiculamente alto comparado ao *QuickMinK*, não considerando o caso mais básico *(k = 1)*, visto que é necessário percorrer o vetor várias vezes, ainda tendo em vista que no último caso, *(k=n)* se é necessário ordenar o vetor inteiro para se obter o resultado, algo que não acontece no *QuickMinK*, nunca será necessário ordenar o vetor inteiro, reduzindo muito o tempo de execução e não tendo tantas chamadas.