

## Alexandre Aparecido Scrocaro Junior R.A.: 2135485

### 5) Tabelas de comparações em segundos

**k = 1**

	<b>n = 1000</b>	<b>n = 10000</b>	<b>n = 100000</b>	<b>n = 500000</b>	<b>n = 1000000</b>
<b>SelectionMink</b>	0,000008	0,000059	0,000497	0,002561	0,005404
<b>HeapMink</b>	0,000040	0,000394	0,003810	0,019049	0,039549
<b>QuickMink</b>	0,000011	0,000380	0,003504	0,013255	0,030546

**k = n/3**

	<b>n = 1000</b>	<b>n = 10000</b>	<b>n = 100000</b>	<b>n = 500000</b>	<b>n = 1000000</b>
<b>SelectionMink</b>	0,000869	0,084305	8,458281		
<b>HeapMink</b>	0,002464	0,233805	24,144880		
<b>QuickMink</b>	0,000047	0,000370	0,006674		

**k = n/2**

	<b>n = 1000</b>	<b>n = 10000</b>	<b>n = 100000</b>	<b>n = 500000</b>	<b>n = 1000000</b>
<b>SelectionMink</b>	0,001175	0,122305	11,443846		
<b>HeapMink</b>	0,003624	0,338231	31,340267		
<b>QuickMink</b>	0,000041	0,000540	0,004515		

**k = n**

	<b>n = 1000</b>	<b>n = 10000</b>	<b>n = 100000</b>	<b>n = 500000</b>	<b>n = 1000000</b>
<b>SelectionMink</b>	0,001606	0,154358	16,435518		
<b>HeapMink</b>	0,004724	0,421650	45,628944		
<b>QuickMink</b>	0,000027	0,000245	0,002383		

## 6) Padrão nos resultados

Sim, quanto maior os números/quantidade de números, leva-se mais tempo para compilar. Além disso, ao analisar a tabela, pode-se dizer que o quick sort se mostrou o melhor algoritmo a ser usado nesse problema, porque, no geral, foi o mais rápido (salvo no caso  $k=1$ , mas não ficou muito atrás do mais rápido).