

Algoritmos escolhidos:

- (i) Bubble Sort - Interativo
- (ii) Bubble Sort - Recursivo
- (iii) Quick Sort
- (iiii) Merge Sort

Tempo de execução dos algoritmos:

	Tamanho dos Vetores				
	1000	10000	100000	1000000	10000000
Bubble sort - interative	0.001017	0.113144	17.037.096		
Bubble sort - interative	0.000989	0.132087	17.066.283		
Bubble sort - interative	0.000958	0.130007	17.096.538		
Bubble sort - interative - Média	0.000988	0.125079	17.066.639		
Bubble sort - recursive	0.001316	0.130061	20.908.288		
Bubble sort - recursive	0.002099	0.139455	19.398.866		
Bubble sort - recursive	0.001047	0.091155	18.383.887		
Bubble sort - recursive - Média	0.001487	0.120223	19.563.368		
Quick sort	0.000087	0.000799	0.009836	0.107875	
Quick sort	0.000066	0.000867	0.009528	0.104778	
Quick sort	0.000083	0.000839	0.009533	0.110559	
Quick sort - Média	0.000078	0.000835	0.009632	0.107737	
Merge sort	0.000077	0.000966	0.011708	0.120192	
Merge sort	0.000064	0.000890	0.010715	0.130785	
Merge sort	0.000068	0.000838	0.011434	0.131969	
Merge sort - Média	0.000069	0.000898	0.011285	0.127648	
Tempo dado em segundos					

Tempo médio de execução dos algoritmos:

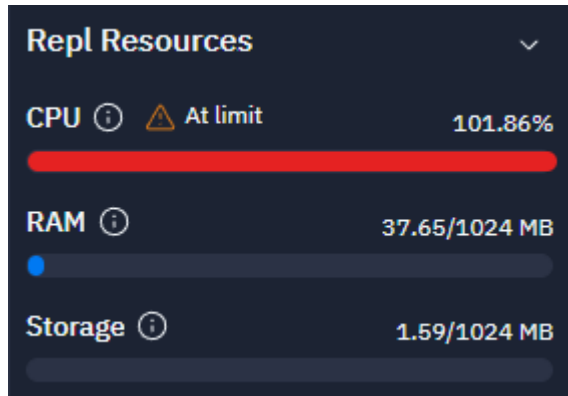
	1000	10000	100000	1000000	10000000
Bubble sort - interative - Média	0.000988	0.125079	17.066.639		
Bubble sort - recursive - Média	0.001487	0.120223	19.563.368		
Quick sort - Média	0.000078	0.000835	0.009632	0.107737	
Merge sort - Média	0.000069	0.000898	0.011285	0.127648	

Nota-se que em alguns casos o tempo de execução se encontra em branco. Isso se deve ao fato dos algoritmos de ordenação terem tido problemas de execução para vetores dos respectivos tamanhos. Em alguns casos, como nos algoritmos Quick Sort e Merge Sort, o algoritmo executa e encerra quase que imediatamente retornando o seguinte erro:

signal: segmentation fault (core dumped) no Replit.com.

```
>_ Console x Shell x +
❯ make -s
❯ ./main
Digite o tamanho do vetor: 10000000
signal: segmentation fault (core dumped)
❯
```

Já nos casos dos algoritmos de ordenação Alg (Bubble Sort neste caso), além de não funcionar também para vetores de 1000000(1 milhão), este apresenta um retorno diferente. Neste caso, o programa executa durante mais de 15 minutos (nos quais o CPU atinge valores iguais e superiores a 100%) e o programa é finalizado sem uma resposta. Em alguns outros compiladores é imprimido a informação “Killed”.



## Comparações:

### a) Alg iterativo X Alg recursivo

	1000	10000	100000	1000000	10000000
Bubble sort - interative - Média	0.000988	0.125079	17.066.639		
Bubble sort - recursive - Média	0.001487	0.120223	19.563.368		

Quando comparados os métodos Alg iterativo e Alg recursivo vemos uma diferença pequena, com os testes feitos levando a uma ideia de que o método iterativo teria uma ligeira vantagem sobre o recursivo.

Isso pode ser interpretado como pouco surpreendente, uma vez que a recursividade, apesar de mais elegante e mais compacto sob visão da ciência da computação, tende a operar de forma ligeiramente mais lenta em uma boa parte dos casos.

### b) Merge sort X Quick sort

	1000	10000	100000	1000000	10000000
Quick sort - Média	0.000078	0.000835	0.009632	0.107737	
Merge sort - Média	0.000069	0.000898	0.011285	0.127648	

Na comparação entre Quick e Merge Sort também vemos diferenças bem sutis quanto ao seu tempo de execução. Onde o Quick Sort se mantém à frente na maioria dos casos. Esse resultado também era esperado, haja vista que uma das particularidades mais bem vistas do Quick Sort é justamente sua constância, onde só em poucos casos (bem trabalhosos para esse método em si) ele apresenta o desempenho abaixo do Merge Sort com valores mais significativos.

Entretanto, na grande maioria dos casos o Quick faz jus ao seu nome e tem um ordenamento mais veloz.

c) A versão mais rápida de alg X o mais lento dentre Mergesort e Quicksort

Ou seja:

c) Bubble Sort - Interativo X Merge Sort

	1000	10000	100000	1000000	10000000
Bubble sort - interative - Média	0.000988	0.125079	17.066.639		
Merge sort - Média	0.000069	0.000898	0.011285	0.127648	

Em contrapartida às diferenças de tempo de execução quase imperceptíveis das comparações anteriores, o Merge Sort (mesmo não sendo o mais veloz) apresenta um salto de eficiência gigantesco em relação ao método mais rápido dentre os algoritmos de ordenação Alg.

Aqui também não tivemos muitas surpresas. Isso se dá principalmente pelo fato do Merge Sort (assim como o Quick Sort) utilizar o princípio de divisão e conquista na sua implementação. Enquanto o Bubble Sort, verifica múltiplas vezes valores que já estão em ordem.

Este teste deixa mais que claro qual a melhor escolha para se implementar sempre que possível, principalmente no que diz respeito a interação do usuário, que não deseja ficar muitos segundos ou até mesmo minutos, aguardando por suas informações.