

## Vetor 1

Algoritmo	Tempo (ms)	Iterações	Trocas	Observação
Comb Sort	47,9561	129	22	Tempo alto, poucas trocas
Gnome Sort	0,0111	176	78	Muitas trocas
Bucket Sort	0,2673	20	33	Poucas iterações
Bubble Sort (flag)	0,0142	180	78	Trocas elevadas
Selection Sort	0,0117	190	18	Menos trocas
Cocktail Sort	0,0154	154	78	Desempenho regular

## Vetor 2

Algoritmo	Tempo (ms)	Iterações	Trocas	Observação
Comb Sort	0,0041	110	0	Fez iterações desnecessárias
Gnome Sort	0,0018	20	0	Excelente desempenho
Bucket Sort	0,0068	20	0	Constante e linear
Bubble Sort (flag)	0,0019	19	0	Parou rapidamente
Selection Sort	0,0055	190	0	Não se beneficia do vetor ordenado
Cocktail Sort	0,0024	19	0	Rápido e eficiente

## Vetor 3

Algoritmo	Tempo (ms)	Iterações	Trocas	Observação
Comb Sort	0,0101	129	18	Bom equilíbrio
Gnome Sort	0,0141	400	190	Muitas iterações e trocas
Bucket Sort	0,0158	20	85	Estável
Bubble Sort (flag)	0,0089	190	190	Mesmo numero de iterações e trocas
Selection Sort	0,0082	190	10	Poucas trocas
Cocktail Sort	0,0089	190	190	Similar ao Bubble Sort

### Resumo Geral:

- **Selection Sort** apresentou o menor número de trocas entre todos os algoritmos, destacando-se nos vetores desordenados e invertidos. Seu comportamento previsível e estável o torna eficiente em cenários pequenos ou com pouca variabilidade de dados.
- **Bubble Sort (com flag de parada)** demonstrou o melhor desempenho para vetores já ordenados, interrompendo a execução rapidamente ao detectar que não há mais trocas necessárias.
- **Bucket Sort** se mostrou muito eficiente em número de iterações, mantendo comportamento quase linear. É o mais indicado para conjuntos de dados com distribuição uniforme.
- **Comb Sort** manteve um equilíbrio consistente entre tempo e número de trocas, apresentando desempenho sólido em diferentes situações e tamanhos de entrada.

- **Gnome Sort e o Bubble Sort tradicional foram os menos eficientes em vetores grandes, devido ao alto número de comparações e trocas, o que reforça sua limitação para cenários com maior volume de dados.**
- **Selection Sort — líder em menor número de trocas nos cenários aleatório e invertido; tempo estável, mas faz muitas comparações (não se beneficia quando o vetor já está ordenado). Bom quando custo de troca é mais crítico que comparações.**
- **Cocktail Sort — variação bidirecional do Bubble: pode reduzir passadas quando há elementos fora de lugar nas duas extremidades. Em ordenado empata com Bubble (flag); em aleatório/invertido tende a ser semelhante ao Bubble.**