

Relatório Henrique Amorim Soares: QuickSort com Variação na Escolha do Pivô

1. Funcionamento de Cada Estratégia de Escolha do Pivô

- **Primeiro elemento:** O pivô é sempre o primeiro elemento do subarray. É uma estratégia simples, mas pode gerar partições desbalanceadas em arrays ordenados ou quase ordenados.
- **Último elemento:** O pivô é sempre o último elemento do subarray. Também é simples, mas sofre dos mesmos problemas da escolha pelo primeiro elemento.
- **Pivô aleatório:** O pivô é escolhido aleatoriamente dentro do subarray, o que reduz significativamente a chance de ocorrer o pior caso.
- **Mediana de três:** O pivô é a mediana entre o primeiro, o elemento do meio e o último elemento do subarray. Essa abordagem melhora o balanceamento das partições.

2. Desempenho Observado

Os testes foram realizados com arrays de 100, 1.000 e 10.000 elementos, considerando três cenários distintos: ordenado, quase ordenado e aleatório. O tempo de execução foi medido em milissegundos para cada estratégia.

Resultados dos testes

Tamanho do array: 100

- **Tipo: Ordenado**
 - Primeiro elemento: 0,920100 ms
 - Último elemento: 0,238300 ms
 - Aleatório: 0,255100 ms
 - Mediana de três: 0,018500 ms
- **Tipo: Quase Ordenado**
 - Primeiro elemento: 0,030400 ms
 - Último elemento: 0,042600 ms
 - Aleatório: 0,088200 ms
 - Mediana de três: 0,017600 ms
- **Tipo: Aleatório**
 - Primeiro elemento: 0,017100 ms
 - Último elemento: 0,018400 ms
 - Aleatório: 0,076600 ms
 - Mediana de três: 0,021800 ms

Tamanho do array: 1000

- **Tipo: Ordenado**
 - Primeiro elemento: 1,221900 ms
 - Último elemento: 1,288500 ms
 - Aleatório: 0,624900 ms
 - Mediana de três: 0,332800 ms
- **Tipo: Quase Ordenado**
 - Primeiro elemento: 0,529700 ms
 - Último elemento: 0,355800 ms
 - Aleatório: 0,085800 ms
 - Mediana de três: 0,044900 ms
- **Tipo: Aleatório**
 - Primeiro elemento: 0,066800 ms
 - Último elemento: 0,035600 ms
 - Aleatório: 0,099200 ms
 - Mediana de três: 0,038200 ms

Tamanho do array: 10000

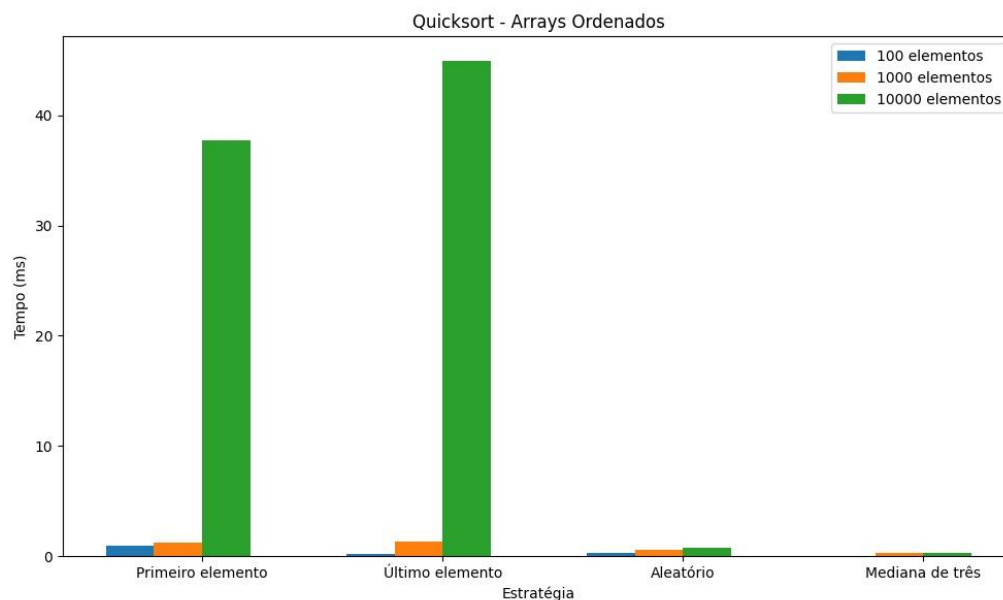
- **Tipo: Ordenado**
 - Primeiro elemento: 37,688000 ms
 - Último elemento: 44,904700 ms
 - Aleatório: 0,811800 ms
 - Mediana de três: 0,315300 ms
- **Tipo: Quase Ordenado**
 - Primeiro elemento: 0,704200 ms
 - Último elemento: 0,900600 ms
 - Aleatório: 0,811100 ms
 - Mediana de três: 0,303400 ms

- **Tipo: Aleatório**
 - Primeiro elemento: 0,503500 ms
 - Último elemento: 0,483800 ms
 - Aleatório: 0,968300 ms
 - Mediana de três: 0,511000 ms

Gráfico comparativo dos resultados (tabela)

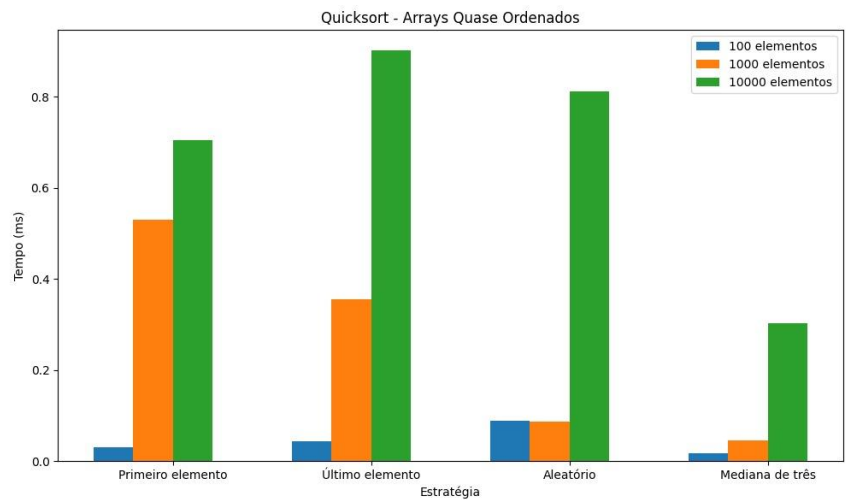
Arrays Ordenados

Estratégia	100 elementos	1000 elementos	10000 elementos
Primeiro elemento	0,920 ms	1,222 ms	37,688 ms
Último elemento	0,238 ms	1,289 ms	44,905 ms
Aleatório	0,255 ms	0,625 ms	0,812 ms
Mediana de três	0,019 ms	0,333 ms	0,315 ms



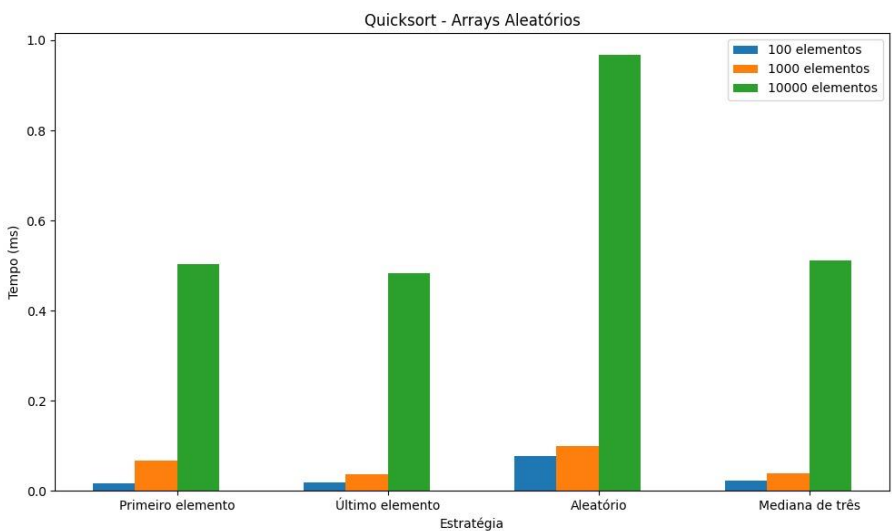
Arrays Quase Ordenados

Estratégia	100 elementos	1000 elementos	10000 elementos
Primeiro elemento	0,030 ms	0,530 ms	0,704 ms
Último elemento	0,043 ms	0,356 ms	0,901 ms
Aleatório	0,088 ms	0,086 ms	0,811 ms
Mediana de três	0,018 ms	0,045 ms	0,303 ms



Arrays Aleatórios

Estratégia	100 elementos	1000 elementos	10000 elementos
Primeiro elemento	0,017 ms	0,067 ms	0,504 ms
Último elemento	0,018 ms	0,036 ms	0,484 ms
Aleatório	0,077 ms	0,099 ms	0,968 ms
Mediana de três	0,022 ms	0,038 ms	0,511 ms



4. Conclusão

A escolha do pivô no QuickSort é super importante pro desempenho, ainda mais em listas grandes ou com padrões. Nossos testes mostraram que usar o primeiro ou o último elemento como pivô é muito ruim se a lista já estiver ordenada. O algoritmo fica lentíssimo e as divisões ficam todas desequilibradas.

Por outro lado, o pivô aleatório e, principalmente, a mediana de três, funcionam muito bem em qualquer situação. Eles conseguem manter o QuickSort rápido porque evitam o pior caso.

Em listas quase ordenadas ou totalmente aleatórias, todas as estratégias até que vão bem, mas a mediana de três é a melhor opção pra garantir que as partições fiquem equilibradas, sendo a mais indicada para uso geral. O pivô aleatório também é uma boa pedida, diminuindo os riscos.

A dica é: evite pivôs fixos em aplicações de verdade, a não ser que você saiba exatamente como são seus dados. O ideal é ir de mediana de três ou pivô aleatório. E claro, sempre teste! Com diferentes dados e tamanhos, pra ter certeza que o QuickSort vai performar do jeito certo.