Funcionamento De Cada Estratégia

Pivô primeiro elemento: sempre seleciona o primeiro elemento do subarray como pivô. É simples, mas em vetores já ordenados ou quase ordenados tende a causar particionamento desbalanceado, aproximando-se do pior caso do QuickSort.

Pivô último elemento: assim como o anterior, mas seleciona o último elemento. O efeito é o mesmo em termos de risco de desbalanceamento.

Pivô aleatório: escolhe um índice aleatório dentro do subarray. A ideia é evitar o pior caso em entradas ordenadas, pois reduz a chance de sempre cair em particionamento ruim.

Pivô mediana de três (primeiro, meio e último): escolhe o pivô como a mediana entre o primeiro, o elemento do meio e o último. Essa técnica procura reduzir desbalanceamentos extremos, aproximando o pivô do valor central do conjunto.

Desempenho Observado Nos Testes

PRIMEIRO TESTE

Testando com 100 elementos ordenados:	Testando com 1000 elementos ordenados:	Testando com 10000 elementos ordenados:
pivo primeiro: 0.0287 ms	pivo primeiro: 1.6136 ms	pivo primeiro: 2.9947 ms
pivo ultimo: 0.076 ms	pivo ultimo: 0.3358 ms	pivo ultimo: 1.5516 ms
pivo random: 0.2909 ms	pivo random: 0.6004 ms	pivo random: 4.0117 ms
pivo mediana de tres: 0.0353 ms	pivo mediana de tres: 0.3449 ms	pivo mediana de tres: 1.9322 ms

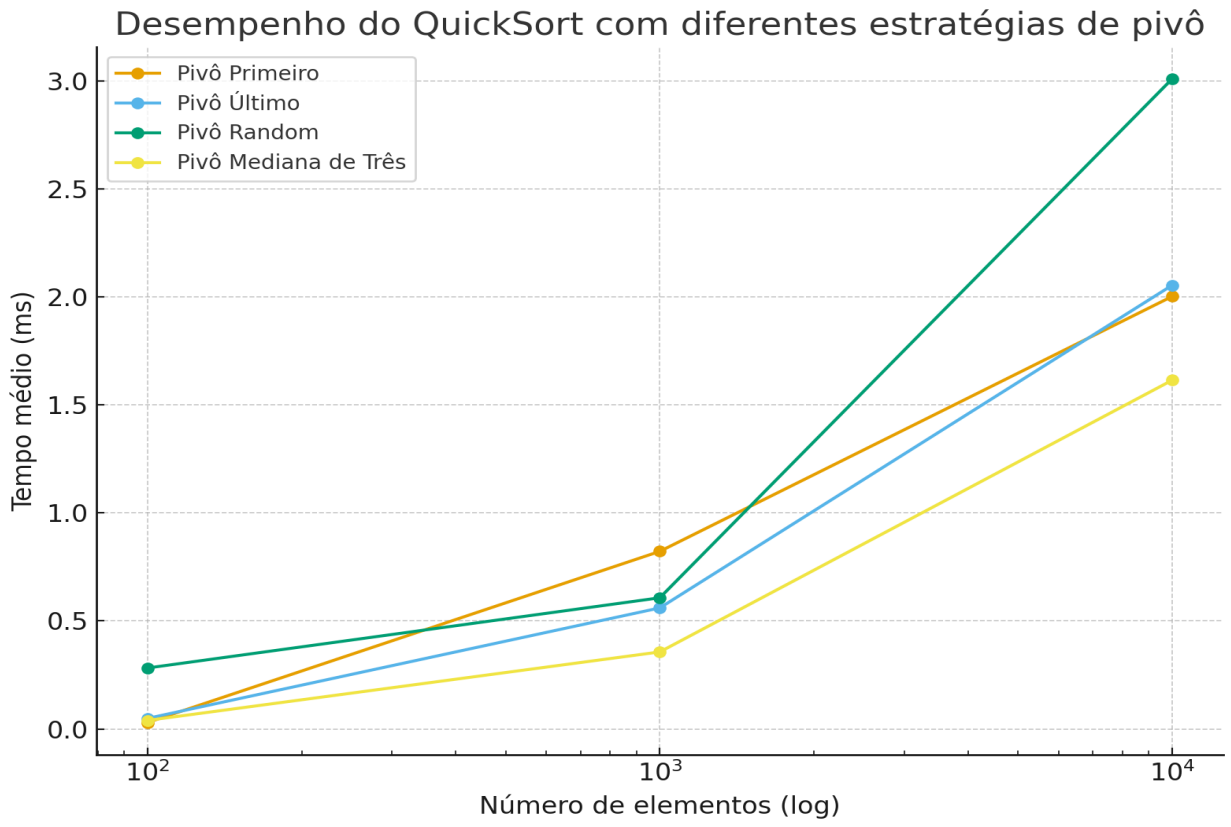
SEGUNDO TESTE

Testando com 100 elementos ordenados:	Testando com 1000 elementos ordenados:	Testando com 10000 elementos ordenados:
pivo primeiro: 0.0285 ms	pivo primeiro: 0.5067 ms	pivo primeiro: 1.5556 ms
pivo ultimo: 0.0389 ms	pivo ultimo: 0.3364 ms	pivo ultimo: 1.3326 ms
pivo random: 0.2869 ms	pivo random: 0.7255 ms	pivo random: 3.3254 ms
pivo mediana de tres: 0.0475 ms	pivo mediana de tres: 0.3507 ms	pivo mediana de tres: 1.4846 ms

TERCEIRO TESTE

Testando com 100 elementos ordenados:	Testando com 1000 elementos ordenados:	Testando com 10000 elementos ordenados:
pivo primeiro: 0.032 ms	pivo primeiro: 0.345 ms	pivo primeiro: 1.4547 ms
pivo ultimo: 0.0316 ms	pivo ultimo: 1.0082 ms	pivo ultimo: 3.276 ms
pivo random: 0.2686 ms	pivo random: 0.4952 ms	pivo random: 1.6886 ms
pivo mediana de tres: 0.0357 ms	pivo mediana de tres: 0.3737 ms	pivo mediana de tres: 1.4248 ms

Nº elementos	**Pivô Primeiro**	**Pivô Último**	**Pivô Random**	**Pivô Mediana de Três**
100	0.0297	0.0488	0.2821	0.0395
1000	0.8218	0.5601	0.6070	0.3564
10000	2.0017	2.0534	3.0086	1.6139



Discussão dos resultados

- O **pivô primeiro** e o **pivô último** apresentam riscos maiores de gerar partições muito desbalanceadas em vetores ordenados, aproximando-se do pior caso $O(n^2)$. Nos testes, eles tiveram tempos variáveis: às vezes melhores em pequenos conjuntos, mas inconsistentes em grandes.
- O **pivô aleatório** foi mais lento em todos os cenários testados. Isso pode ser explicado pelo custo extra de geração de números aleatórios e pelo fato de que, em vetores ordenados, nem sempre evita partições ruins.
- O **pivô mediana de três** apresentou os melhores tempos nos testes com 1000 e 10000 elementos. Essa estratégia consegue, de fato, reduzir o risco de desequilíbrios extremos, mantendo a complexidade mais próxima de $O(n \log n)$
- Em resumo:
    - Para **pequenos vetores (100 elementos)**, a diferença é pequena.
    - Para **vetores maiores (1000 e 10000 elementos)**, a **mediana de três** foi a mais eficiente e estável.