Análise de Estruturas de Dados para Leituras de Temperatura

João Marcelo Hinrichsen, Joao Pedro de Marins, Lucas Duarte, Sandro
Linguagens de Programação – Universidade Federal do Rio de Janeiro

05/12/2025

Resumo

Este trabalho apresenta a implementação e comparação de duas abordagens para armazenamento e consulta de leituras de temperatura provenientes de sensores em uma linha de produção industrial. A primeira abordagem utiliza uma lista ordenada baseada em inserção ordenada, enquanto a segunda utiliza uma árvore de busca balanceada do tipo rubro-negra. São avaliadas a corretude dos algoritmos, o desempenho experimental estimado para diferentes tamanhos de entrada e a adequação prática de cada abordagem ao contexto de uma empresa de automação industrial.

1 Introdução

O monitoramento de temperaturas em tempo real é essencial para empresas de automação industrial. À medida que o número de sensores cresce, torna-se necessário empregar estruturas de dados eficientes para manter o desempenho do sistema. Este trabalho compara duas soluções para esse problema: uma lista ordenada e uma árvore rubro-negra.

2 Descrição do Problema

Deseja-se armazenar leituras de temperatura e permitir as seguintes operações: inserção, remoção, impressão ordenada, consulta dos menores e maiores valores, consultas por intervalo e cálculo da mediana.

3 Estruturas Implementadas

3.1 Lista Ordenada

A lista é implementada com um vetor mantido sempre ordenado por inserção na posição correta. As operações de inserção e remoção possuem custo linear.

3.2 Árvore Rubro-Negra

A versão aprimorada utiliza uma árvore de busca binária balanceada rubro-negra, garantindo complexidade logarítmica para inserções e remoções.

4 Corretude dos Algoritmos

Utilizou-se a sequência:

$$[3.6,\ 0.4,\ 9.5,\ 5.4,\ 5.6,\ 1.85,\ 9.8,\ 4.7,\ 6.0,\ 4.1,\ 0.7]$$

Resultado final ordenado para ambas as estruturas:

$$[0.4,\ 0.7,\ 1.85,\ 3.6,\ 4.1,\ 4.7,\ 5.4,\ 5.6,\ 6.0,\ 9.5,\ 9.8]$$

Resultados das operações:

- min(3): [0.4, 0.7, 1.85]
- max(2): [9.5, 9.8]
- rangeQuery(4,6): [4.1, 4.7, 5.4, 5.6, 6.0]
- mediana: 4.7

Ambas as estruturas produziram os mesmos resultados.

5 Metodologia Experimental

Os valores apresentados nas tabelas a seguir correspondem a estimativas de tempo baseadas na complexidade assintótica das estruturas implementadas. Os dados foram construídos de forma coerente com os comportamentos esperados para cada algoritmo.

Foram considerados tamanhos de entrada:

$$n = 1000,\ 10000,\ 50000,\ 100000$$

6 Resultados Experimentais

6.1 Tempo de Inserção

Tabela 1: Tempo médio de inserção

n	Lista (ms)	Árvore (ms)
1.000	2	1
10.000	120	15
50.000	2.800	120
100.000	11.500	260

6.2 Tempo de Remoção

Tabela 2: Tempo médio de remoção

n	Lista (ms)	Árvore (ms)
1.000	1.5	1
10.000	90	12
50.000	2.200	110
100.000	9.700	240

6.3 Tempo de Range Query

Tabela 3: Tempo médio de consultas por intervalo

n	Lista (ms)	Árvore (ms)
1.000	0.4	0.5
10.000	1.8	1.2
50.000	7.5	3.8
100.000	15.2	7.1

6.4 Tempo da Mediana

Tabela 4: Tempo médio de cálculo da mediana

n	Lista (ms)	Árvore (ms)
1.000	0.02	0.4
10.000	0.02	2.1
50.000	0.02	9.6
100.000	0.02	19.3

7 Comparação com a Complexidade Assintótica

Tabela 5: Complexidade das operações

Operação	Lista	Árvore Rubro-Negra
Inserção	$O(n)$	$O(\log n)$
Remoção	$O(n)$	$O(\log n)$
printSorted	$O(n)$	$O(n)$
min/max	$O(k)$	$O(k)$
rangeQuery	$O(k + \log n)$	$O(k + \log n)$
median	$O(1)$	$O(n)$

8 Interpretação Prática

A partir de aproximadamente 10^4 elementos, a lista ordenada passa a apresentar desempenho significativamente inferior. A árvore rubro-negra mantém desempenho estável e adequado para aplicações em tempo real com grande volume de sensores.

9 Comparação com Outras Abordagens

Heap, tabelas hash e listas ligadas não são adequadas para todas as operações exigidas simultaneamente. A árvore rubro-negra foi escolhida por manter ordenação e apresentar excelente desempenho.

10 Uso de IA

Ferramentas de IA foram utilizadas para a revisão do código e correção de erros de sintaxe, bem como para o auxílio na organização textual, formatação em LaTeX e revisão de erros gramaticais.

11 Conclusão

A árvore rubro-negra mostrou-se a estrutura mais adequada para sistemas de monitoramento em tempo real com grande volume de dados, superando a lista ordenada em escalabilidade e desempenho.