



Augusto Ortigoso Barbosa

**Trabalho Fila e Pilha - Implementação e Análise
Estrutura de Dados I
Profa. Daniele Carvalho Oliveira**

**Monte Carmelo - MG
2024**

Sumário

1	Benchmark de Execução	3
2	Discussão das Hipóteses - Estrutura de Dados Fila	4
2.1	Inserção na Fila	4
2.2	Remoção na Fila	4
2.3	Impressão na Fila	4
3	Discussão das Hipóteses - Estrutura de Dados Pilha	5
3.1	Inserção na Fila	5
3.2	Remoção na Fila	5
3.3	Impressão na Fila	5
4	Perguntas	6
5	Conclusão Geral	7

1 Benchmark de Execução

Tabela de Tempo de Execução em Clocks - Filha Estática (VisualStudio Code)																								
Operação	Quantidade de Elementos																							
	1000				3000				5000				10000				25000							
No. Da Bateria	1	2	3	M	1	2	3	M	1	2	3	M	1	2	3	M	1	2	3	M				
Inserção	1	0	0	0,33333	0	1	0	0,33333	0	1	1	0,66667	0	0	0	0	1	1	0	0,66667				
Remoção	0	0	0	0	2	0	1	1	1	0	0	0,33333	1	2	1	1,33333	0	1	1	0,66667				
Listagem	22	36	34	30,6667	115	113	122	116,667	154	153	150	152,333	308	317	353	326	730	974	959	887,667				
Tabela de Tempo de Execução em Clocks - Filha Dinâmica (VisualStudio Code)																								
Operação	Quantidade de Elementos																							
	1000				3000				5000				10000				25000							
No. Da Bateria	1	2	3	M	1	2	3	M	1	2	3	M	1	2	3	M	1	2	3	M				
Inserção	0	0	2	0,66667	0	0	0	0	1	1	1	1	0	0	1	0,33333	2	1	1	1,33333				
Remoção	0	0	0	0	0	1	0	0,33333	0	0	1	0,33333	1	0	0	0,33333	1	1	1	1				
Listagem	26	39	178	81	87	96	111	98	152	185	156	164,333	299	339	324	320,667	1002	1090	996	1029,33				
Tabela de Tempo de Execução em Clocks - Filha Estática (VisualStudio Code)																								
Operação	Quantidade de Elementos																							
	1000				3000				5000				10000				25000							
No. Da Bateria	1	2	3	M	1	2	3	M	1	2	3	M	1	2	3	M	1	2	3	M				
Inserção	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
Remoção	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
Listagem	36	35	42	37,6667	87	108	94	96,3333	194	197	228	206,333	320	357	301	326	788	845	967	866,667				
Tabela de Tempo de Execução em Clocks - Filha Dinâmica (VisualStudio Code)																								
Operação	Quantidade de Elementos																							
	1000				3000				5000				10000				25000							
No. Da Bateria	1	2	3	M	1	2	3	M	1	2	3	M	1	2	3	M	1	2	3	M				
Inserção	0	0	1	0,33333	0	1	1	0,66667	1	0	1	0,66667	1	1	0	0,66667	1	1	1	1				
Remoção	0	0	0	0	1	0	0	0,33333	0	0	0	0	0	0	1	0,33333	1	0	1	0,66667				
Listagem	25	32	29	28,6667	86	88	92	88,6667	157	152	153	154	297	340	320	319	747	737	992	825,333				

2 Discussão das Hipóteses - Estrutura de Dados Fila

2.1 Inserção na Fila

Hipótese: O tempo de execução da inserção em uma fila sequencial é menor do que em uma fila dinâmica.

Análise: De acordo com os resultados obtidos, o tempo médio de inserção na fila estática é geralmente menor em relação à fila dinâmica, principalmente em conjuntos menores (1000 a 3000 elementos). Isso ocorre porque a fila estática usa um bloco contíguo de memória, o que facilita o acesso e a inserção rápida, enquanto a fila dinâmica exige alocação de novos nós e ponteiros, o que implica um custo adicional em tempo. Portanto, essa hipótese é confirmada.

2.2 Remoção na Fila

Hipótese: O tempo de execução da remoção em uma fila dinâmica é menor do que em uma fila sequencial.

Análise: Os dados mostram tempos de remoção muito próximos para ambas as implementações, com a fila dinâmica geralmente tendo uma vantagem leve ou tempos zerados em algumas baterias. A fila dinâmica é mais eficiente para remoção em filas grandes, pois só precisa alterar ponteiros ao invés de realocar elementos. Assim, a hipótese é parcialmente confirmada, pois a diferença é mais significativa em filas maiores.

2.3 Impressão na Fila

Hipótese: O tempo de execução da operação de impressão na fila sequencial é menor do que na fila dinâmica.

Análise: Os resultados mostram que a fila estática apresenta, em média, um tempo de execução menor para impressão do que a fila dinâmica. Isso ocorre porque, na fila estática, os elementos estão contiguamente alocados, o que facilita a leitura sequencial. Já na fila dinâmica, a impressão exige percorrer ponteiros, o que pode aumentar o tempo de execução. Logo, essa hipótese é confirmada.

3 Discussão das Hipóteses - Estrutura de Dados Pilha

3.1 Inserção na Fila

Hipótese: O tempo de execução da inserção em uma pilha sequencial é menor do que em uma pilha dinâmica.

Análise: Os tempos médios indicam que a pilha estática é ligeiramente mais rápida para inserção. Na pilha estática, o acesso à memória é direto, enquanto a pilha dinâmica requer criação de nós e ajuste de ponteiros, o que eleva o custo em tempo. Assim, essa hipótese é confirmada, especialmente em entradas menores.

3.2 Remoção na Fila

Hipótese: O tempo de execução da remoção em uma pilha sequencial é menor do que em uma pilha dinâmica.

Análise: Os tempos de remoção na pilha sequencial são consistentemente baixos ou nulos, independentemente do tamanho da entrada. A remoção em uma pilha sequencial implica apenas o decremento do topo, enquanto, na pilha dinâmica, há necessidade de desalocar o nó removido. Portanto, essa hipótese também é confirmada.

3.3 Impressão na Fila

Hipótese: O tempo de execução da operação de impressão na pilha sequencial é menor do que na pilha dinâmica.

Análise: A pilha estática apresentou tempos médios de impressão menores em comparação com a pilha dinâmica, principalmente em conjuntos menores de dados. Como na fila, o bloco contíguo de memória facilita o acesso direto aos elementos, enquanto na pilha dinâmica é necessário percorrer cada nó. Dessa forma, essa hipótese é confirmada.

4 Perguntas

1. Em quais quantidades de elementos e em quais operações a implementação usando memória sequencial teve o menor tempo?

- **Para a Fila Estática:**

- **Inserção:** 1000 elementos, com tempo médio de 0.3333 clocks.
- **Remoção:** Para todas as quantidades de elementos, o tempo foi 0 clocks.
- **Listagem:** O menor tempo foi com 1000 elementos, média de 30.6667 clocks.

- **Para a Pilha Estática:**

- **Inserção e Remoção:** Ambas as operações tiveram o tempo de 0 clocks para todas as quantidades de elementos.
- **Listagem:** Menor tempo com 1000 elementos, média de 37.6667 clocks.

2. Em quais quantidades de elementos e em quais operações a implementação usando memória dinâmica teve o menor tempo?

- **Para a Fila Dinâmica:**

- **Inserção:** O menor tempo foi com 1000 elementos, média de 0.6667 clocks.
- **Remoção:** 0 clocks para 1000 e 3000 elementos.
- **Listagem:** O menor tempo foi com 1000 elementos, média de 81 clocks.

- **Para a Pilha Dinâmica:**

- **Inserção:** O menor tempo foi com 1000 elementos, média de 0.3333 clocks.
- **Remoção:** 0 clocks para todas as quantidades de elementos testadas.
- **Listagem:** O menor tempo foi com 1000 elementos, média de 28.6667 clocks.

5 Conclusão Geral

Os dados de benchmark mostram que a implementação em memória sequencial é geralmente mais rápida para operações de inserção, remoção e impressão tanto em filas quanto em pilhas, especialmente para conjuntos de dados menores. A vantagem da implementação sequencial é sua simplicidade e acesso direto à memória contígua. No entanto, para grandes conjuntos de dados, a memória dinâmica oferece flexibilidade superior e melhor uso de recursos, embora isso venha com um custo de tempo adicional para operações que envolvem alocação e realocação de memória.

Em resumo:

- Para tarefas com alto volume de inserção e remoção, a memória dinâmica pode ser vantajosa em termos de flexibilidade, apesar de um leve aumento no tempo de execução.
- Para operações intensivas de leitura e impressão, a memória sequencial geralmente oferece melhor desempenho devido ao acesso direto aos elementos.