

Trabalho da M1 de Estruturas de Dados:

Comparação entre Listas

Maria Julia Lamim Severino, Nicole Migliorini Magagnin

1. Comparativo

Para a análise da complexidade nas operações de inserção, foi estabelecido como critério as quantidades de comparações, as trocas e as inserções do elemento feitas ao longo das operações. Nossas fórmulas de cálculo foram baseadas em: troca dentro de troca($n*n$), troca simples(n) e demais inserções, remoções e chamadas de função (+1).

Foi analisado que na inserção no início, temos desvantagens se implementarmos na lista estática, uma vez que precisamos fazer uma troca para deslocarmos n vezes todos os elementos para a sua próxima posição e assim inserir o novo elemento. Por outro lado, a inserção na lista encadeada é a mais simples por precisar apenas de uma comparação e duas inserções.

Durante a análise da inserção no fim, percebemos que a lista encadeada se torna mais complexa por necessitar que passemos por toda lista até encontrarmos o último elemento, além disso foram necessárias uma comparação e duas inserções. Em contrapartida, a lista estática fica em vantagem, visto que é possível encontrarmos facilmente o último elemento e assim poder inserir o novo logo após e em segundo lugar, temos a lista duplamente encadeada, pela qual já temos acesso ao fim da lista, necessitando apenas de uma comparação e duas inserções.

Para inserir em qualquer posição, podemos notar que a vantagem é da lista estática, dado que para essa operação não há necessidade de utilizar duas inserções. Diferente das listas encadeada e duplamente encadeada, que ficam em desvantagem já que além de comparação e troca são necessárias as duas inserções.

Table 1. Comparativo entre operações de inserção

	Lista Estática	Lista Encadeada	Lista Encadeada	Duplamente
InserirInício	$n+2$	3		4
InserirFim	2	$n+3$		3
InserirPosição	$n+2$	$n+2$		$n+3$

Quanto a remoção, pudemos notar uma vantagem das listas encadeadas na remoção do início, uma vez que as mesmas tem acesso direto ao início da lista, necessitando poucas trocas para tal e uma vantagem da lista estática na remoção do fim, já que a mesma só precisa reduzir o número de elementos, seguida pela lista duplamente encadeada, que tem acesso ao fim da lista a todo momento.

Na remoção por posição, a vantagem também manteve essa ordem, devido a facilidade de alcance do fim da lista pelas listas estática e duplamente encadeada.

Table 2. Comparativo entre as operações de remoção

	Lista Estática	Lista Encadeada	Lista Encadeada Duplamente
Remove Início	$n+1$	2	2
Remove Fim	1	$n+3$	3
Remove Posição	$n+1$	$n+3$	$n+2$

Para as ordenações, notamos que o Bubblesort torna-se semelhante em todas as listas, tendo igual complexidade e uma complexidade menor que o Quicksort, esse, possui menor complexidade na lista estática, devido ao fato de não precisar contar a posição dos elementos na lista, chamando um número menor de funções, mesmo que tenhamos utilizado a função de troca “swap” não presente nas outras listas.

Por termos acesso ao final da lista e aos elementos anteriores, o Quicksort torna-se mais simples na lista duplamente encadeada que na lista simplesmente encadeada, pois necessita de uma chamada de função a menos, entretanto, a diferença é pequena.

Table 3. Comparativo entre Bubblesort e QuickSort

	Lista Estática	Lista Encadeada	Lista Encadeada Duplamente
Bubblesort	$n^2 + 1$	$n^2 + 1$	$n^2 + 1$
Quicksort	$n^3 + 6$	$n^3 + 8$	$n^3 + 7$

2. Conclusões

Após as análises dos resultados concluímos que nas operações de inserção a lista estática necessita o deslocamento da maioria de seus elementos, ocasionando um maior custo. Já para as listas encadeadas, é necessário passar por toda a lista até encontrar a posição desejada.

Nas operações de remoção, a lista estática e duplamente encadeada possui vantagens por poderem acessar seu fim e início com mais facilidade. A lista simplesmente encadeada, é uma boa escolha apenas para a remoção no início.

Por fim, nas operações de ordenação, foi possível analisar que ambas as listas possuem a mesma complexidade na operação de Bubblesort e quando utilizado a função Quicksort a lista estática é de menor custo.

De modo geral, a lista estática apresenta uma estrutura mais básica e vantajosa para a maioria dos casos, porém para programas complexos a lista duplamente encadeada é a melhor estrutura para as operações citadas, uma vez que não necessita de um tamanho predeterminado e assim não se torna uma estrutura limitada.