# Trabalho da M1 de Estruturas de Dados: Comparação entre Listas

**Diogo Marchi, Leonardo Grando e Patrick de Luca**

# 1. Comparativo

Table 1. Comparativo entre operações de inserção

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Lista Estática** | **Lista Encadeada** | **Lista Duplamente Endadeada** |
| **InsereInício** | (n) | 1 | 1 |
| **InsereFim** | 1 | (n) | 1 |
| **InserePosição** | (n-(pos-1)) | Pos-1 | Pos-1 |

A desvantagem é da lista estática na inserção é que dependendo da posição em que o elemento é inserido, se cria a necessidade de realocar todo o resto da lista, já nas listas encadeada e duplamente encadeada não se tem essa necessidade. A única necessidade passa a ser o fato de que o elemento anterior precisa apontar para seu novo sucessor. Contudo, na operação de se inserir no fim, a lista encadeada fica em desvantagem, pois tem a necessidade de percorrer toda ela, o que não acontece nas outras duas listas

Table 2. Comparativo entre as operações de remoção

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Lista Estática** | **Lista Encadeada** | **Lista Duplamente Encadeada** |
| **Remove Início** | (n-1) | 1 | 1 |
| **Remove Fim** | 1 | (n) | 1 |
| **Remove Posição** | n-pos | Pos-1 | Pos-1 |

Novamente a desvantagem da lista estática é o fato de se ter a necessidade de realocar os outros elementos da lista, mas a vantagem é que sempre se sabe as posições de cada item. A lista encadeada fica para trás novamente com a necessidade de percorrer todos os elementos para se chegar no fim dela e assim remover seu último elemento. Já a duplamente encadeada tem a vantagem de se percorrer a lista em duas direções.

Table 3. Comparativo entre Bubblesort e QuickSort

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Lista Estática** | **Lista Encadeada** | **Lista Duplamente Encadeada** |
| **Bubblesort** | ((n²-n)/2) | (n²-n) | (n²-n) |
| **Quicksort** | (n-1)\*n | [(n-1)\*n)]+(n+n/2+1) | [(n-1)\*n)]+(n+n/2+1) |
|  |  |  |  |

## Ja aqui, pelo método bubblesort a lista estática ganha uma certa vantagem por ser mais barata que a encadeada e duplamente encadeada. Pelo método de quicksort ocorre a mesma coisa, por ter que percorrer a lista atrás do ultimo elemento e do elemento que se encontra na metade da lista, ele acaba custando mais.

# 2. Conclusões

O trabalho serviu para esclarecer a função e momento de uso de cada tipo de lista. Além de nos mostrar que velocidade não é tudo, tem de haver custo benefício. Cada lista tem suas vantagens e desvantagens, momentos que devem e não devem ser usadas. Em geral a lista duplamente encadeada parece ser a com melhor custo benefício, apesar de suas desvantagens, é a que melhor utiliza a alocação de memória sem ocupar espaço desnecessário e também a que tem melhor mobilidade, não precisando realocar seus elementos e também podendo percorrer as duas direções da lista.