

1 Introdução

2 Lista Lineares

2.1 Introdução

Dentre as estruturas de dados não primitivas, as listas lineares são as de manipulação mais simples. Uma lista linear agrupa informações referentes a um conjunto de elementos que, de alguma forma, se relacionam entre si, podendo ser inúmeros os tipos de dados que podem ser descritos por listas lineares.

Uma lista linear também pode ser entendida como uma tabela ou um vetor, ela é composta por um conjunto de $n \geq 0$ nós tais que suas propriedades estruturam decorrem unicamente da posição relativa dos nós dentro da sequência linear. Os comportamentos mais comuns das listas — e que na verdade estão presente na maioria das estruturas de dados — são de remoção, adição e busca. Entretanto, existem tipos de listas com variações específicas desses comportamentos que podem ser entendidas como estruturas de dados únicas pela forma como se comportam, e como seu uso único é muito particular, porém útil, não somente na computação. Algumas dessas “listas especiais” são:

- **Fila:** Quando a inserção dos elementos é feita de um lado da lista, mas a remoção é feita no outro lado da lista, e cada uma dessas ações é exclusivamente executada no seu lado correspondente, seguindo o padrão *FIFO*.
- **Pilha:** A inserção e a deleção de elementos são feitos no mesmo lado da lista, e são exclusivamente feitos desse lado, proporcionando assim o uso do princípio *LIFO*.
- **Deque:** É uma abreviação de *double ended queue* e ela tem como comportamento, a possibilidade de adição e deleção de elemento dos dois lados da lista, e apenas nas extremidades, assim como as estruturas especiais anteriormente citadas.

O armazenamento dos elementos presentes dentro de uma lista linear pode ser classificado de acordo com a posição relativa na memória de dois nós consecutivos na lista. Quando o armazenamento é sempre contínuo, podemos dizer que é uma *alocação sequencial de memória*, por outro lado, quando isso não é feito, podemos dizer que temos uma *alocação encadeada*. A escolha de

um tipo ou do outro depende essencialmente das operações que são executadas sobre a lista, do número de listas envolvidas e como as características particulares dessas listas.

2.2 Alocação Sequencial

A maneira mais simples de se manter uma lista linear na memória de um computador é colocar seus nós em posições contínuas. Como a implementação da alocação sequencial em linguagens de alto nível é geralmente realizada com reserva prévia de memória, a inserção e remoção de nós não ocorrem de fato. Normalmente são usadas variáveis para "definir" os limites da memória, o que nos faz concluir que essa alocação pode ser considerada *estática*.

Ele é particularmente atraente no caso de filas e pilhas já que suas operações básicas podem ser implementadas de forma bastante eficiente. Entretanto, é muito oneroso computacionalmente falando, essas operações em questão de memória, por isso, um estudo da sua necessidade deve ser feito antes da sua implementação.

2.3 Pilhas e Filas

O armazenamento sequencial de listas é empregado quando as estruturas ao longo do tempo, sofrem poucas remoções e inserções. Em casos particulares de listas, esse armazenamento é também empregado. Nesse caso, a situação favorável é aquela em que inserções e remoções não acarretam movimentação de nós, o que ocorre se os elementos a serem inseridos e removidos estão em posições especiais, com a primeira ou a última posição.

2.4 Alocação Encadeada

A implementação de operações realizadas em listas com alocações sequenciais é fraca, principalmente quando há utilização concomitante de mais de duas listas, pois torna a gerência de memória mais complexa, com isso, justifica-se o uso de alocação encadeada, que também pode ser chamada de *alocação dinâmica*. Os nós de uma lista são alocados e desalocados conforme a necessidade, e com isso, eles se encontram aleatoriamente dispostos na memória e são interligados por ponteiros, que indicam a posição do próximo elemento da tabela. É necessário o acréscimo de um campo a cada nó, justamente o que indica o endereço do próximo nó da lista. Apesar dessa forma necessitar de mais memória por utilizar mais um campo informacional, é mais conveniente quando o problema inclui o tratamento de mais de uma lista, isso se

aplica tanto à gerência do armazenamento quanto às operações propriamente ditas envolvidas.

2.5 Lista Lineares em Alocação Encadeada

2.5.1 Listas Simplesmente Encadeadas

Qualquer estrutura que seja armazenada em alocação encadeada requer o uso de um tipo de ponteiro que indique o endereço de seu primeiro nó. O percurso da lista é feito a partir dos ponteiros, seguindo consecutivamente os endereços fornecidos a cada um deles, até chegar ao objetivo.

Esse tipo de estrutura, pode apresentar erros, como por exemplo, considerar durante um algoritmo de busca, que o nó cabeça não possui um ponteiro indicado que ele é o primeiro, necessitando dentro da busca, casos de teste especiais para verificar que ele é o primeiro elemento. Isso pode ser resolvido com uma modificação da estrutura que provê uma classificação especial aos primeiros e último nós, o que facilita a busca deles dentro do sistema.

2.5.2 Pilhas e Filas

Como em casos particulares, modificações são necessárias para a eficiência da implementação de operações em pilhas e filas. No caso das **pilhas**, não é necessário a presença do nó cabeça, pois temos que o fim da inserção e o fim da início da exclusão é sempre feito a partir do elemento que está no topo da pilha, facilitando assim o algoritmo. No caso das **filas**, temos duas variáveis que apontam para o primeiro e o último elementos inseridos dentro da estrutura, em que são respectivamente a saída e a entrada da fila.

2.5.3 Lista Duplamente Encadeada

A lista duplamente encadeada possui comportamento similar ao da lista simplesmente encadeada, com a única diferença de que um ponteiro extra é adicionado, e que esse mesmo indica o elemento anterior o qual acabou de ser citado e/ou analisado.

Outro fator que deve ser analisado, e que é parecido com o da lista simplesmente encadeada, é a necessidade da presença de uma variável especial para indicar o início da lista, nesse caso a necessidade de indicar o último elemento da lista também se faz presente, evitando assim buscas infinitas e circulares.

Aplicação: Ordenação Topológica A ordenação topológica é um problema muito presente dentro da matéria de Teoria dos Grafos, e esse tipo de

algoritmo pode ser resolvido com o uso de uma lista duplamente encadeada. Sua importância se deve ao seu potencial de todas as vezes que o problema abordado envolve uma ordem parcial — que é entendida como um conjunto S que é a relação entre os objetos de S , temos o entendimento da seguinte maneira:

3 Árvores

3.1 Definição e Representação Básica

Temos que uma árvore enraizada, que é mais comumente conhecida como somente árvore, é um conjunto finito de elementos denominados de *nós* ou *vértices*, tais que

- A árvore é dita vazia na ausência de elementos.
- Existe um nó especial chamado raiz. Os restantes constituem um único conjunto vazio, ou são divididos em $m \geq 1$ conjuntos disjuntos não vazios.