
UFES

ESTRUTURA DE DADOS

WikiHow

Autor
JOÃO PEDRO MILLI
BIANCARDI

November 26, 2020



Contents

1	Proposta e introdução	2
2	Sobre a WikiHow	2
3	Estrutura	3
4	Funcionamento	4
5	Função Caminho	5
6	Considerações finais	8

1 Proposta e introdução

A proposta do presente trabalho é aplicar os conhecimentos obtidos na disciplina, criando um programa em C utilizando conceitos de Estruturas de Dados. Esse relatório abordará o método utilizado para a criação do programa, sua estrutura e funcionamento, métodos utilizados para aplicações e funções mais complexas.

2 Sobre a WikiHow

A WikiHow é um programa em C que processa informações a respeito de determinados tópicos, montando estruturas que são chamadas Páginas, cada página carrega informações sobre o tópico no qual ela aborda, assim como as pessoas que contribuíram para sua criação e referências a outras páginas, tudo isso feito através de arquivos de texto. Especificações sobre sua estrutura serão abordadas a seguir.

3 Estrutura

A WikiHow é estruturada em 8 TADs, sendo 4 tipos diferentes e 4 listas para cada tipo. Cada TAD tem seu arquivo .c e de cabeçalho associado, com funções e especificações para sua criação, manipulação e destruição, especificações mais detalhadas de cada TAD e função estarão presentes nos arquivos de cabeçalhos. Os TADs utilizados são:

Tipo Editor: TAD para criação e manipulação dos contribuintes da página;

Tipo Contribuição: TAD para criação e manipulação das contribuições que serão inseridas em determinada página;

Tipo Link: TAD para criação e manipulação das referências a outras páginas que serão inseridas em determinada página;

Tipo Página: TAD para criação e manipulação da página propriamente dita, que será formada pelos TADs anteriores;

Tipo ListaEditor: TAD para criação e manipulação de listas formadas por tipos Editor;

Tipo ListaCont: TAD para criação e manipulação de listas formadas por tipos Contribuição;

Tipo ListaLink: TAD para criação e manipulação de listas formadas por tipos Link;

Tipo ListaPag: TAD para criação e manipulação de listas formadas por tipos Página;

As listas foram todas implementadas utilizando a estrutura de listas simplesmente encadeadas com sentinela, ou seja, cada objeto está inserido em uma célula que contém o endereço de memória para a próxima célula da lista contento o próximo objeto, cada lista é controlado por uma sentinela que contém os endereços para o início e o final da fila. Especificações sobre seu funcionamento serão abordadas a seguir.

4 Funcionamento

A WikiHow sempre receberá um argumento de entrada que será um arquivo de texto entrada.txt, esse arquivo conterá os comandos que deverão ser realizados pelo programa, cada comando deverá ter os argumentos necessários para sua execução, por exemplo, para criar um editor é necessário informar o seu respectivo nome, EX:

INSEREEDITOR Joao

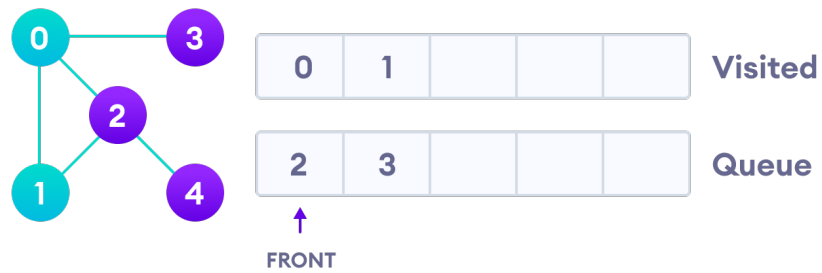
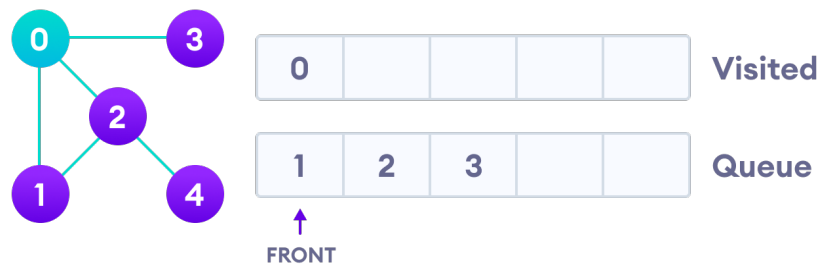
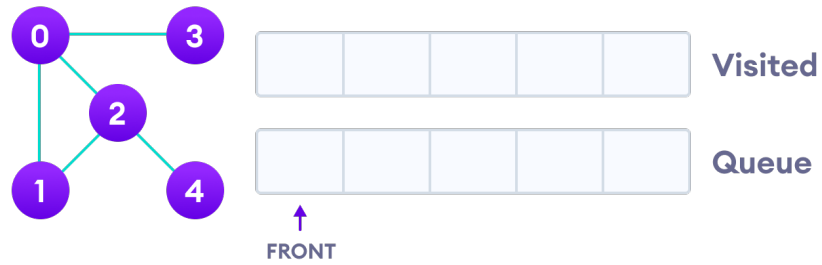
A especificação de todos os comandos com seus respectivos argumentos podem ser encontrados no PDF disponibilizado pela professora, a entrada deverá sempre terminar em FIM, indicando a finalização do programa

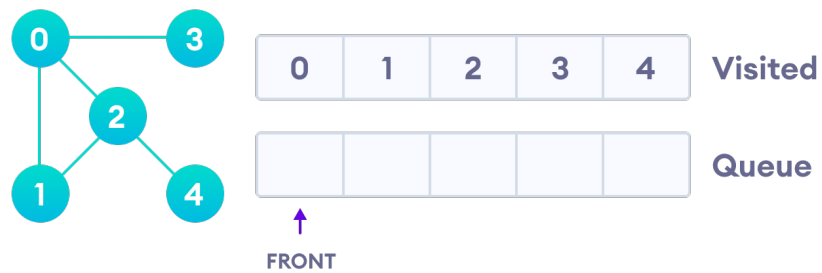
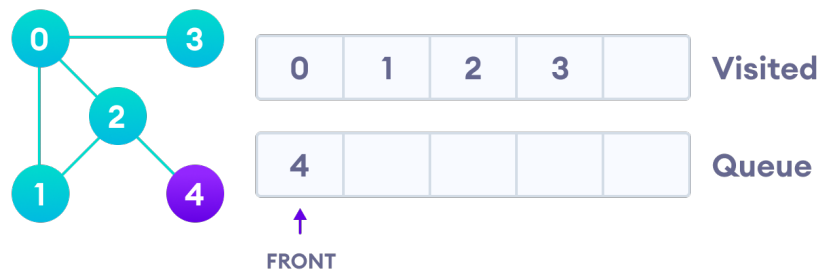
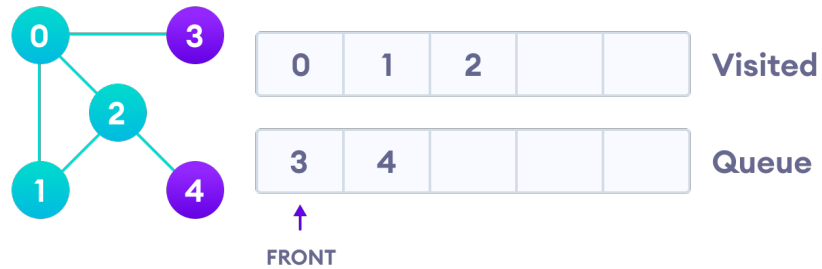
As contribuições em texto serão inseridas a partir de arquivos .txt que devem estar presentes no mesmo diretório da WikiHow. Contribuições poderão então ser criadas através de comandos na entrada, especificando o nome do arquivo de texto e a Página na qual essa contribuição será inserida, é valido lembrar que cada contribuição necessita de um Editor responsável. Ao final do processo, é possível obter as informações de todas ou de uma determinada página criada, para isso o cliente deverá inserir o comando IMPRIMEWIKED ou IMPRIMEPAGINA especificando a página de interesse. Será criado um arquivo .txt no diretório da WikiHow para cada página contendo todas as suas informações: título, contribuintes, contribuições e links para outras páginas. É importante lembrar que cada objeto é criado pelo próprio usuário através de comandos na entrada, e que o programa liberará toda a memória associada após sua execução

Na execução do programa, será criado um arquivo de texto com o nome de log.txt, esse arquivo conterá informações sobre eventuais erros que ocorrerem durante a execução do programa, como por exemplo a tentativa de destruir um objeto inexistente, a remoção de uma contribuição que não foi feita pelo editor, inserção de contribuição em uma página inexistente, etc. Caso não haja nenhum erro na execução, o arquivo estará vazio.

5 Função Caminho

A função mais complexa presente no trabalho foi a função CAMINHO, que checa se existe caminho indireto entre duas páginas. Sua aplicação é complexa, por isso foi necessário a pesquisa de algoritmos capazes de realizar o trabalho. Após muita pesquisa foi encontrado um algoritmo chamado de BFS Graph Algorithm (Breadth first search), um algoritmo recursivo para procurar todos os vértices de um grafo ou estrutura de dados do tipo árvore, sua implementação coloca cada vértice de um grafo em duas categorias: Visitado e não Visitado, o propósito do algoritmo é marcar cada vértice como visitado enquanto evita loops infinitos. O algoritmo começa colocando algum vértice do grafo em uma queue (vértices marcados para serem analisados), após análise, ele retira o vértice em análise da queue e o adiciona em uma lista de vértices já visitados, após isso o algoritmo obtém todos os vértices adjacentes ao vértice em análise e, caso ainda não estejam na lista de visitados ou na queue, são adicionados a lista da queue, o algoritmo volta a analisar o primeiro item da queue e se repete até que a lista da queue esteja vazia.





Pseudocódigo:

```

cria uma queue Q
marca v como visitada e coloca v em Q
enquanto Q é não vazia
    remove o header u de Q
    marca e coloca na queue todas (não visitadas) vizinhas de u
  
```


6 Considerações finais

Acredito que todas as propostas do trabalho foram satisfeitas, as melhorias para futuras atualizações poderão conter novas funções e uma reformulação na funcionalidade de erros, o programa abre e fecha o arquivo log.txt toda vez que um erro é detectado, essa ação é custosa e ineficiente, uma alternativa seria criar um TAD de erros, salvar os erros acumulados em uma lista e abrir o arquivo de log.txt apenas uma vez.