

Implementação de Pilha e Fila

Mariana Bastos dos Santos

Junho 2019

1 Introdução

Na computação, existem diversos mecanismos de organização de dados para atender aos diferentes requisitos de processamento. Esses mecanismos são as estruturas de dados, as quais definem como distribuir e relacionar os dados disponíveis, tornando mais eficientes os algoritmos que manipulam esses dados. Sendo assim, este trabalho tem como objetivo implementar duas diferentes estruturas de dados: pilha e fila. Utilizando linguagem c++ para o desenvolvimento, são criadas as classes pilha e fila, sendo que a primeira irá empilhar as letras de uma palavra e irá desempilhar essas letras e a segunda irá enfileirar diferentes palavras e as desenfileirá na sequência.

2 Trabalhos Relacionados

[1] propõe um algoritmo de escalonamento de tarefas baseado em algoritmos de clustering fuzzy. Para isso, foi combinado o algoritmo de clusterização fuzzy c-means baseado em kernel e o algoritmo FIFO (First-In-First-Out) aprimorado para projetar um novo algoritmo de planejamento. No algoritmo de clusterização fuzzy c-means baseado em kernel, usamos função de base radial (RBF) como função kernel. Os resultados do experimento mostram que o algoritmo proposto encurta o tempo de execução das tarefas e aumenta a utilização dos recursos.

3 Conceitos Fundamentais

3.1 Fila

Também conhecida como FIFO (First-In-First-Out), é uma estrutura de dados onde o primeiro elemento é processado primeiro e o mais novo elemento é processado depois. Por exemplo, numa bilheteria as pessoas chegam, pegam os ingressos e vão embora. As pessoas entram numa fila para serem atendidas de maneira organizada. A pessoa que entrar na fila primeiro receberá o ingresso primeiro e sairá da fila. A pessoa que entrar na fila depois, receberá o ingresso depois que a pessoa na frente dele. Sendo assim, a pessoa que entrar na fila por último será o último a receber os ingressos.

3.2 Pilha

Também conhecida como LIFO (Last-In-Last-Out), é uma estrutura de dados onde o primeiro elemento é processado por último e o último elemento é processado primeiro. Por exemplo, em um tubo onde se armazenam bolas, diferentes tipos de bolas são colocadas. A bola que entrar no tubo por último, será tirada primeiro. A bola que estiver entre a primeira colocada e a última colocada, só será retirada quando a última for retirada. Sendo assim a bola que entrar primeiro no tubo será a última a sair.

4 Implementação

Neste trabalho foram implementadas as estruturas de dados, pila e fila. Para a implementação de ambas foi utilizada a linguagem C++.

Na figura 1 é possível observar o diagrama da implementação da classe pilha. Os atributos da classe são as letras de uma palavra representada por `s[tamanho]` e o topo da pilha representado pela variável inteira `topo`. Uma palavra foi lida, por exemplo "Mariana", na sequência foi chamado o método inicializa da classe observado na figura 2. Cada caractere da palavra "Mariana" foi inserido um a um no método empilha observado na figura 3, onde cada caractere da palavra é empilhado. Na sequência foi chamado o método desempilha, observado na figura 4, onde cada letra da palavra "Mariana" é desempilhada, formando então uma nova palavra: "anairaM".

Na figura 1 é possível observar o diagrama da implementação da classe fila. Os atributos da classe são o primeiro elemento da fila representado por

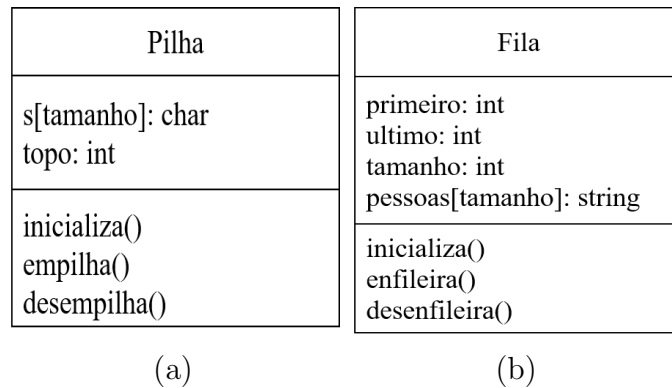


Figura 1: (a): Diagrama da classe pilha (b): Diagrama da classe fila

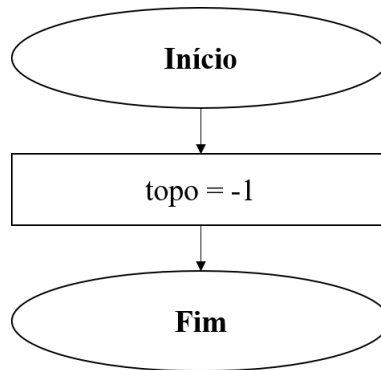


Figura 2: Fluxograma do método inicializa da classe pilha

primeiro, último elemento da fila representado por ultimo, tamanho da fila representado por tamanho e as pessoas que serão elementos da fila representado pelo vetor pessoas[tamanho]. Inicialmente foi populado um vetor com n nomes de pessoas que entrariam na fila. Na sequência foi chamado o método inicializa da classe observado na figura 5. Cada um dos nomes populados no vetor de n nomes foi enfileirado pelo método enfileira observado na figura 5. Na sequência foi chamado o método desenfileira observado na figura 5 no qual cada uma das pessoas são retiradas da fila pela ordem correta do algoritmo, a primeira que entra é a primeira que sai e a última que entra é a última que sai.

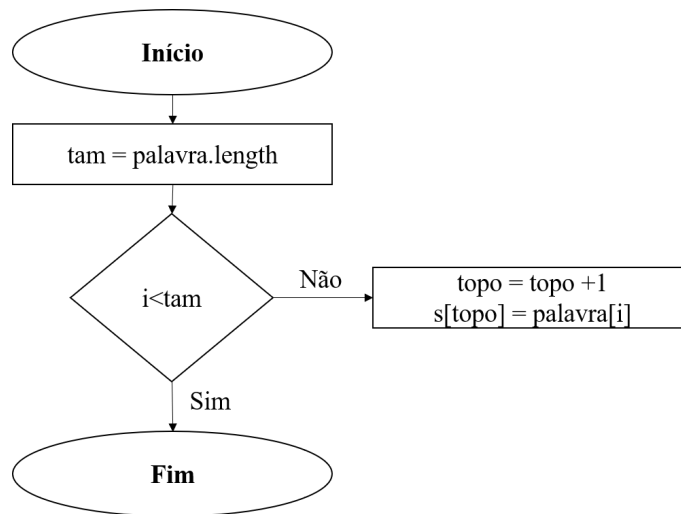


Figura 3: Fluxograma do método empilha da classe pilha

5 Experimentos e Resultados

O experimento realizado nesse trabalho para a classe pilha foi imprimir cada um dos elementos inseridos no método empilhar e verificar se a posição a qual foram inseridos correspondia com o resultado esperado para o método. Na sequência foi impresso a posição do topo da pilha e a respectiva letra que estava nessa posição para verificar se a pilha estava funcionando de maneira correta. Em seguida, foi chamada a função desempilha e então listada a ordem a qual foram desempilhadas cada uma das letras inseridas na pilha. No resultado obtido e observado na figura 6 é possível verificar que houve resultados esperados, o algoritmo empilhou de forma correta as palavras, apresentou o topo com o valor correto e letra correta, e por fim, o método desempilha seguiu a ordem esperada para o algoritmo de pilha, primeiro a entrar último a sair e último a entrar primeiro a sair.

O experimento realizado nesse trabalho para a classe fila foi imprimir cada um dos elementos a serem enfileirados e suas respectivas posições, a fim de verificar se o método estava funcionando da maneira correta. Na sequência foi impresso o elemento e sua respectiva posição e a ação de desenfileirar, para saber se o método estava sendo chamado corretamente. Em seguida, foi impressa o nome da pessoa desenfileirada. No resultado obtido e observado na figura 7 é possível verificar que houve resultados esperados, o algoritmo

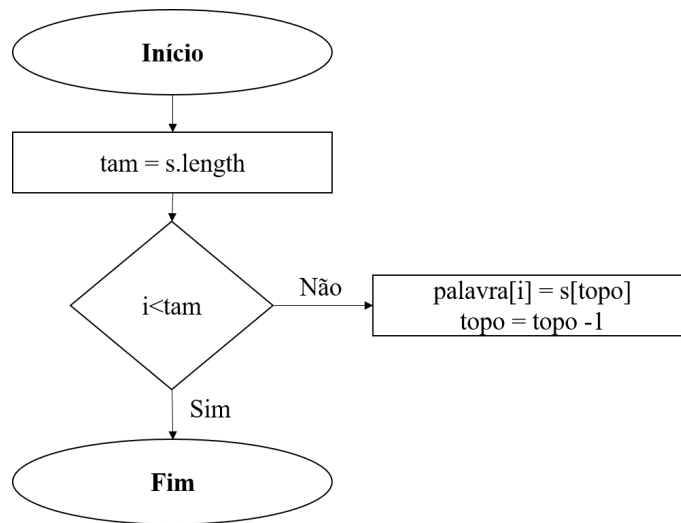


Figura 4: Fluxograma do método desempilha da classe pilha

seguir com o método de enfileirar de forma correta e o mesmo ocorreu para o método de desenfileirar.

6 Conclusão

Esse trabalho propôs a implementação das estruturas de dados fila e pilha. Nos experimentos foram alcançados resultados esperados para o objetivo do trabalho, no qual cada um dos algoritmos apresentaram comportamento esperado.

Referências

- [1] Jian Li, Tinghuai Ma, Meili Tang, Wenhai Shen, and Yuanfeng Jin. Improved fifo scheduling algorithm based on fuzzy clustering in cloud computing. *Information*, 8:25, 2017.

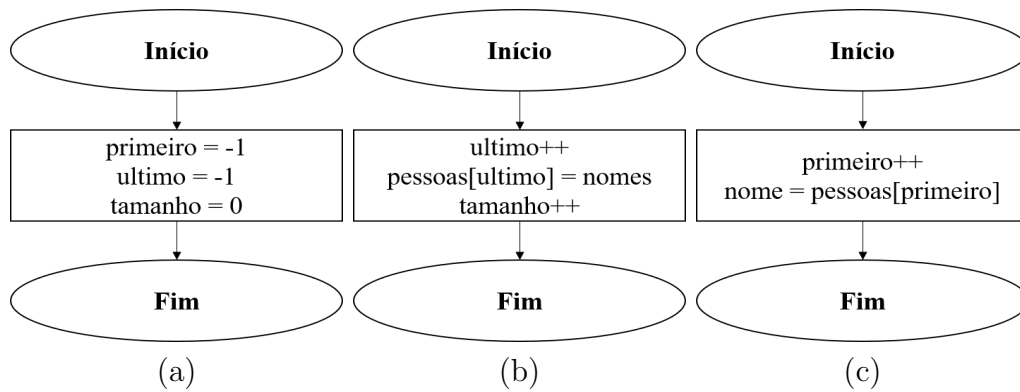


Figura 5: (a): Fluxograma do método inicializa da classe fila. (b): Fluxograma do método empilha da classe fila. (c) Fluxograma do método desempilha da classe fila.

```

PILHA
#####
Empilha: M
Empilha: a
Empilha: r
Empilha: i
Empilha: a
Empilha: n
Empilha: a
Topo: 6
Letra Topo: a
Desempilha: anairaM
#####

```

Figura 6: Resultado da classe pilha

Fila	Fila[0] = Mariana
#####	Desenfileirando...
Enfileirando...	Nome = Mariana
Fila[0] = Mariana	Fila[1] = Joao
	Desenfileirando...
Enfileirando...	Nome = Joao
Fila[1] = Joao	Fila[2] = Pedro
	Desenfileirando...
Enfileirando...	Nome = Pedro
Fila[2] = Pedro	Fila[3] = Maria
	Desenfileirando...
Enfileirando...	Nome = Maria
Fila[3] = Maria	Fila[4] = Juliana
	Desenfileirando...
Enfileirando...	Nome = Juliana
Fila[4] = Juliana	Fila[5] = Sofia
	Desenfileirando...
Enfileirando...	Nome = Sofia
Fila[5] = Sofia	Fila[6] = Enzo
	Desenfileirando...
Enfileirando...	Nome = Enzo
Fila[6] = Enzo	Fila[7] = Pietra
	Desenfileirando...
Enfileirando...	Nome = Pietra
Fila[7] = Pietra	#####
	Pressione qualquer tecla para continuar. . .

Figura 7: Resultado da classe fila