

Desenvolvimento de estrutura de dados – Pilha e Fila

Autores

Antonio Pegorelli Neto – RA:119101-4

Introdução

Com o avanço da inteligência artificial e soluções de big data, é importante o uso de estruturas eficientes de armazenamento de dados que possam diminuir o esforço computacional e o uso de memória dos dispositivos. O objetivo deste trabalho é implementar e testar as classes de estrutura de dados para filas e pilhas em C++.

As estruturas de fila e pilha serão implementadas em classes e seus dados armazenados em vetores com capacidade máxima de 500 números inteiros.

Teoria

Definido pelo livro Algoritmos (3ª edição - Cormen, T; Leiserson, C; Rivest R e Stein, C), em uma pilha o último elemento inserido é o primeiro a ser retirado (ou LIFO – Last-In, First-Out), no caso da fila o primeiro elemento a ser retirado é o primeiro que foi colocado na estrutura (ou FIFO – First-In, First-out).

Proposta

A implementação da classe pilha envolve a criação do próprio vetor com tamanho 500 onde serão colocados os dados, uma variável com o tamanho atual da pilha e suas funções de interface:

- push – Para inserir um dado novo na pilha;
- pop – Para remover um dado da pilha;
- top – Para verificar o dado do topo da pilha;
- print – Para imprimir na tela os dados da pilha.

UML da classe pilha:

Stack
values[500]: int stackSize: int
pop() push(int value) top() printStack()

No caso da fila, também é criado um vetor Para a fila é necessário as funções:

- enqueue – Para inserir um dado novo na fila;
- dequeue – Para retirar um dado da fila;
- frente – Para verificar o primeiro dado da fila;
- print – Para imprimir na tela a lista.

UML da classe fila:

Queue
values[500]: int start: int finish: int queueSize: int
dequeue() enqueue(int value) frente() printQueue()

Experimentos e Resultados

Para realizar o teste das estruturas de dados, fizemos um breve código que cria uma estrutura de teste de cada tipo com valores inteiros aleatórios e chama algumas vezes cada método das duas classes, indicando na tela o que foi feito. A resposta dos métodos foi satisfatória, realizando todos os processos e corretamente impedindo que a pilha ou fila admitissem dados acima da capacidade ou retornar dados teoricamente já excluídos caso as estruturas estiverem vazias.

Trabalhos Correlatos

Apesar da sua simples implementação, há uma variedade no método de armazenamento dos dados para as estruturas estudadas em substituição ao vetor, incluindo listas ligadas, que não apresentam um limite específico para a quantidade de dados a ser armazenada.

Conclusão

As estruturas de pilha e fila são simples e eficientes soluções para armazenamento de dados, que podem ser implementadas com baixo poder computacional e dificilmente apresentam problemas se não for ultrapassado o limite de dados pré-estabelecido. O próximo passo para esse trabalho deve ser a mudança do armazenamento dos dados em listas ligadas, que proporcionam maior flexibilidade com o volume de dados armazenados.

Bibliografia

Algoritmos (3ª edição - Cormen, T; Leiserson, C; Rivest R e Stein, C)