INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO NORTE

Exercícios de Estrutura de Dados

1. Descreva o estado da fila (inicialmente vazia) após cada uma das operações a seguir: (R-4.2)

enqueue(5), enqueue(3), dequeue(), enqueue(2), enqueue(8), dequeue(), dequeue(), enqueue(9), enqueue(1), dequeue(), enqueue(7), enqueue(6), dequeue(), dequeue(), enqueue(4), enqueue (), dequeue().
2. Escreva um algoritmo que inverta uma fila. Este algoritmo deve ser executado em tempo linear.

OBS.: Este algoritmo é externo ao TAD Fila e só pode usar os métodos definidos no TAD fila
3. Implemente uma classe Fila usando a classe Vector para armazenar os elementos internamente.
4. A recepcionista de uma clínica para idosos sempre se atrapalha e nunca acerta a ordem correta de atendimento da clínica, que é a ordem de chegada. O administrador, para resolver o problema, resolveu contratar você para desenvolver um programa que “saiba” sempre quem é o próximo a ser chamado. O programa deverá ter o seguinte menu:

Novo paciente na fila
Próximo paciente a ser atendido
Quantidade de pacientes em espera
FIM

O programa deve armazenar o nome do paciente para facilitar o trabalho da recepcionista.
5. Descreva como implementar o TAD Fila usando duas pilhas. Avalie o tempo de execução dos algoritmos enqueue e dequeue.
6. Descreva como implementar o TAD Pilha usando duas filas para armazenar seus elementos. Qual o tempo de execução das operações push e pop neste caso?
7. Descreva o estado do deque (inicialmente vazio) após cada uma das seguintes operações:

insertFirst(3), insertLast(8), insertLast(9), insertFirst(5), removeFirst(), removeLast(), first(), insertLast(7), removeFirst(), last(), removeLast(). (R.43)
8. Implemente um deque usando arrays. Descreva o tempo de execução de cada um dos algoritmos.
9. Implemente um deque usando Vector.
10. Modificar os TADs Pilha, Fila e Deque implementados com arrays para que além das operações básicas de cada um deles, tem uma operação AcessarMenor que retorna o menor valor armazenado. A sua implementação deve ter complexidade $O(1)$. Depois faça uma tabela com os tempos de execução de cada métodos.