

Tipos Abstractos de Dados Filas

Filas são seqüências de elementos com acesso apenas pelas extremidades, a entrada e a saída. Implementam a disciplina *First In First Out (FIFO)*, o primeiro a entrar é o primeiro a sair.

Considerando a operação de sequenciação, ':', colocar um elemento e um seqüência, temos a seguinte definição para o TAD Fila.

$$Fila = (\{filaVazia, elemento:Fila\}, \{cria, insere, retira, topo, vazia?\})$$

22 Implemente o TAD Fila.

- Defina o tipo Fila, baseando a implementação numa lista (simplesmente) ligada.
- Defina as operações internas como uma funções no novo tipo.

Escreva ficheiros de definição de funções (*header*, .h) e ficheiros de implementação das funções (.c).

23 A estrutura fila é muito usadas na implementação de programas de simulação de filas de espera. Por exemplo, as caixas num supermercado, as filas de carros num semáforo, as filas numa cantina, etc.

Utilizando o TAD Fila de inteiros, escreva um programa que comece por preencher uma fila com 200 valores aleatórios (funções `srand` e `rand`) e que de seguida calcule o valor mínimo, o valor máximo e o valor médio dos elementos da fila.

24 Re-implemente o TAD Fila baseando a implementação do tipo *vectores dinâmicos*.

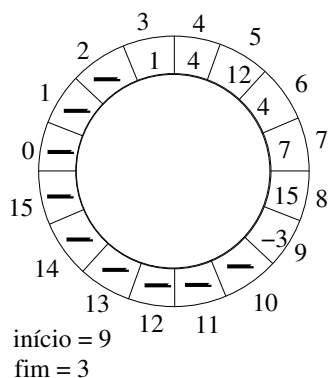
Conceba um conjunto de testes que permita avaliar as vantagens e desvantagens de cada uma das implementações.

25 A implementação de *lista circular simplesmente ligada* é a mais apropriada para o TAD fila.

Re-implemente o TAD Fila baseando a implementação numa *lista circular simplesmente ligada*.

Conceba um conjunto de testes que permita avaliar as vantagens e desvantagens de cada uma das implementações.

26 É possível implementar de forma eficiente o conceito de lista circular utilizando uma representação contígua.



Re-implemente o TAD Fila baseando a implementação nesta nova configuração.

Conceba um conjunto de testes que permita avaliar as vantagens e desvantagens de cada uma das implementações.

27 Deque (double ended queue) – é uma versão generalizada da estrutura de dados de fila que permite inserir e excluir em ambas as extremidades.

Operações do TAD Deque

- criar: criar um novo *deque*;
- inserirInicio: Adiciona um elemento ao início do *deque*;
- inserirFim: Adiciona um elemento no fim *deque*;
- apagarInicio: Apaga um elemento no início do *deque*;
- apagarFim: Apaga um elemento no fim do *deque*;
- verInicio: Obtém o elemento no início do *deque*;
- verFim: Obtém o último elemento do *deque*;
- estaVazia: Verifica se o *deque* está vazio, ou não.

1. Discuta qual será a melhor forma de representar esta nova estrutura. Implemente-a.
2. Implemente as operações que conjuntamente com a nova estrutura definem o TAD *deque*.

28 Considere um dado percurso circular, no qual existem n bombas de gasolina. Pretende-se encontrar o primeiro percurso circular, feito por um dado camião, que visita todas as bombas de gasolina.

Existem dois conjuntos de dados.

- A quantidade de gasolina que todas as bombas de gasolina têm.
- Distância da bomba de gasolina para a próxima bomba de gasolina.

Usando filas como estrutura de dados, calcule o primeiro ponto de onde um camião será capaz de completar o círculo (o camião vai parar em cada bomba de gasolina e tem capacidade infinita). Suponha que para 1 litro de gasolina, o camião pode ir 1 unidade de distância.

Por exemplo, considere que existem 4 bombas de gasolina. A quantidade de combustível e distância entre bombas de gasolina é dada pelos seguintes pares de valores (4, 6), (6, 5), (7, 3) e (4, 5). O primeiro ponto de onde o camião pode fazer um percurso circular é a segunda bomba de gasolina. A saída deve ser «Início = 2^a bomba de gasolina».