

Filas

Prof. Tiago Massoni
Engenharia da Computação
Poli - UPE

Definição

"**Filas** são conjuntos ordenados de dados (i.e. estruturas de dados) nos quais novos elementos podem ser **inseridos** ou elementos pré-existentes podem ser **removidos** em suas duas diferentes extremidades, respectivamente **fim e início**."

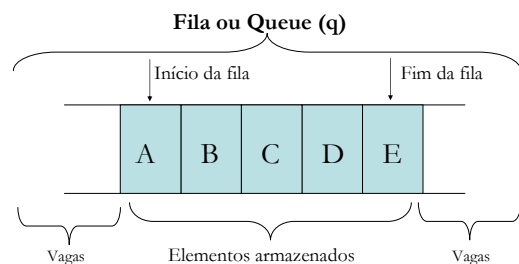
2

Observação

- Pilha garante ordem inversa (Last-In First-Out LIFO)
- Fila garante ordem direta (First-In First-Out FIFO)

3

Intuição



4

Características

- Filas se expandem ou se reduzem ao longo do seu ciclo de vida
- O lado positivo disto: economia de recursos
- O lado menos-positivo disto: não é possível determinar um tamanho de área ideal e portanto prevenir erros de 'estouro' de área de alocação

5

TAD Fila

- Remover ou *dequeue*
- Inserir ou *enqueue*

Dado uma fila q qualquer, as operações acima são especificadas como

$i = q.dequeue()$; leia-se: remova de s e atribua para i

$q.enqueue(i)$; leia-se: insert i na fila q

6

TAD Fila

- Não existe limite (teórico) para inserir dados em q
 - Apenas o limite imposto pelo meio físico de armazenamento
- Antes de remover dado da fila testar se ela é não vazia
 $q.isEmpty()$; leia-se: cheque se a fila q não é fila vazia

7

Estudo de caso: caixas de um banco

- Dado um banco com n pontos de atendimento e uma fila única para clientes
- Condições de atendimento correta:
 1. Pontos de atendimentos livres requisitam um cliente para atender (segundo a ordem de chegada dos mesmos - FIFO)
 2. Ao chegar, clientes entram no final da fila
- Por definição estão erradas as condições:
B ser servido antes que A \Rightarrow viola condição 1
C chegou e fica à frente de B \Rightarrow viola condição 2

8

Outras aplicações

- Fila de impressão
- Fila de pessoas esperando por atendimento de um operador de telemarketing

9

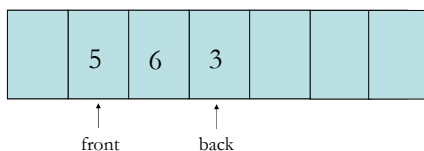
Filas em Java - principais operações

- Construtores
- Fila Vazia
- Inserir
- Remover

Implementação usando listas
ligadas é trivial
Vamos investigar implementação de
filas em array

10

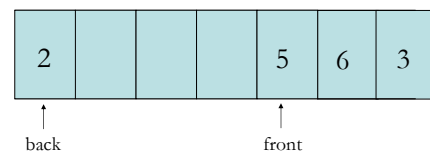
Fila como array



- Enqueue: incrementa back, coloca elemento na posição back
- Dequeue: retorna valor na posição front, incrementa front
- Problema: quando $back == array.length$, não pode inserir, mesmo quando ainda tiver posições livres no array

11

Solução: fila circular



- Fila vazia: front "uma posição à frente" de back
- Fila com um elemento: front = back
- Fila cheia: back "uma posição antes" de front
- Testes adicionais em enqueue
 - Incremento especial

12

Fila circular

```
public class Queue {
    private Object[] array;
    private int size, front, back;
    public Queue(int tam){
        array = new Object[tam];
        makeEmpty();
    }
    ...
    public void makeEmpty(){
        size=0;
        front=0;
        back=-1;
    }
    public void enqueue(Object x) throws
        OverflowException{
        if (isFull()) throw new OverflowException();
        back=increment(back);
        array[back]= x;
        size++;
    }
}
```

13

Fila circular

```
public Object dequeue(){
    if (isEmpty()) return null;
    size--;
    Object item = array[front];
    array[front]=null;
    front=increment(front);
    return item;
}

public int increment(int i){
    i = ++i % array.length;
    return i;
}
...
}
```

14

Exercício

- Implemente uma estrutura de dados fila dinamicamente (usando objetos nós)

15