

## Respostas

1. Em primeiro lugar, podemos transformar o algoritmo deixando apenas os métodos de remoção no início e inserção no início, ou pode-se transformar deixando os métodos de remoção no final e inserção no final. A segunda opção não é recomendada devido a falta de eficiência do algoritmo.
2. Primeira solução – Inserção no final e Remoção no início, ou seja, remoção não eficiente, Segunda solução – Inserção no início e remoção no final (inserção não eficiente)
3. .
  - a. 0
  - b. 1
  - c. 2
  - d. 3
  - e. 4

0	1	2	3	4	0	1	2	3	4	0
---	---	---	---	---	---	---	---	---	---	---

```
class Fila {  
    private int[] array;  
    private int primeiro; // Remove do índice "primeiro".  
    private int ultimo; // Insere no índice "ultimo".
```

```
/**
```

```
 * Construtor da classe.
```

```
*/
```

```
public Fila () {
```

```
    this(6);
```

```
}
```

```
/**
```

```
 * Construtor da classe.
```

```
 * @param tamanho Tamanho da fila.
```

```
*/
```

```
public Fila (int tamanho){  
    array = new int[tamanho+1];  
    primeiro = ultimo = 0;  
}
```

```
/**
```

```
 * Insere um elemento na ultima posicao da fila.
```

```
 * @param x int elemento a ser inserido.
```

```
 * @throws Exception Se a fila estiver cheia.
```

```
*/
```

```
public void inserir(int x) throws Exception {
```

```
    //validar insercao
```

```
    if (((ultimo + 1) % array.length) == primeiro) {  
        throw new Exception("Erro ao inserir!");  
    }
```

```
    array[ultimo] = x;
```

```
    ultimo = (ultimo + 1) % array.length;
```

```
}
```

```
/**
```

```
 * Remove um elemento da primeira posicao da fila e movimenta
```

```
 * os demais elementos para o primeiro da mesma.
```

```
 * @return resp int elemento a ser removido.
```

```
 * @throws Exception Se a fila estiver vazia.
```

```
*/
```

```

public int remover() throws Exception {

    //validar remocao
    if (primeiro == ultimo) {
        throw new Exception("Erro ao remover!");
    }

    int resp = array[primeiro];
    primeiro = (primeiro + 1) % array.length;
    return resp;
}

/**
 * Mostra os array separados por espacos.
 */
public void mostrar (){
    System.out.print("[ ");

    for(int i = primeiro; i != ultimo; i = ((i + 1) % array.length)) {
        System.out.print(array[i] + " ");
    }

    System.out.println("]");
}

public void mostrarRec(){
    System.out.print("[ ");
    mostrarRec(primeiro);
    System.out.println("]");
}

```

```
}
```

```
public void mostrarRec(int i){  
    if(i != ultimo){  
        System.out.print(array[i] + " ");  
        mostrarRec((i + 1) % array.length);  
    }  
}
```

```
/**  
 * Retorna um boolean indicando se a fila esta vazia  
 * @return boolean indicando se a fila esta vazia  
 */  
public boolean isVazia() {  
    return (primeiro == ultimo);  
}  
}
```