

Estruturas de Dados I

com Códigos C++

Prof. Igor Machado Coelho
igor.machado@ime.uerj.br

Departamento de Informática e Ciência da Computação
Universidade do Estado do Rio de Janeiro

IME-04-10820 2015/2



Conteúdo do Curso

Filas

Fila Encadeada

Fila Sequencial



Fila

- ▶ Uma Fila pode ser imaginada como vemos no cotidiano.
- ▶ Na fila do R.U., o indivíduo deve entrar sempre no fim da fila e, o próximo indivíduo a sair será sempre o primeiro da fila.
- ▶ Na prática (R.U.), isso às vezes não acontece! Mas lembrem-se que um computador não é tão tolerante quanto as pessoas em relação a fugir das regras =)



Fila

- ▶ Uma Fila (queue) é uma estrutura de dados linear (assim como as Listas) para armazenamento e busca de informações, consistindo de 3 operações básicas: frente, enfileira (enqueue) e desenfileira (dequeue).
- ▶ Uma Fila pode ser implementada utilizando uma lista linear, mas o acesso aos elementos de uma fila estão restritos a apenas extremidades da lista.
- ▶ Porém, as inserções e remoções só podem acontecer em pontas distintas!
- ▶ Seu comportamento é descrito como FIFO (first-in first-out), ou seja, o primeiro elemento a entrar será o primeiro a sair.
- ▶ Assim como as listas, para o T.A.D. Fila, estudaremos duas formas distintas de implementação: Sequencial e Encadeada.



Fila Encadeada – Definição T.A.D.

As Filas Encadeadas utilizam nós para armazenar as informações.

```
class NoFila1 {  
    char valor;  
    NoFila1* prox;  
};  
class FilaEnc1 {  
    public:  
    NoFila1* inicio;  
    int N; // numero de elementos atualmente na fila  
  
    void constroi(){...}  
    void destroi(){...}  
  
    char frente(){...}  
    void enfileirar(char valor){...}  
    char desenfileirar(){...}  
};
```

Qual a complexidade de espaço?



Fila Encadeada – Utilização

Vamos testar a Fila Encadeada para aprender na prática!

```
int main(){
    FilaEnc1 F;
    F.constroi();
    F.enfileirar('A');
    F.enfileirar('B');
    F.enfileirar('C');
    printf("%c\n", F.frente());
    printf("%c\n", F.desenfileirar());
    F.desenfileirar('D');
    while(F.N > 0)
        printf("%c\n", F.desenfileirar());
    F.destroi();

    return 0;
}
```

Quais são as impressões em tela?



Fila Encadeada – Construção e Destruição

```
class FilaEnc1 {  
    ...  
    void constroi(){  
        N = 0;  
        inicio = NULL;  
    }  
  
    void destroi(){  
        while(inicio != NULL){  
            NoFila1* p = inicio->prox;  
            delete inicio;  
            inicio = p;  
        }  
    }  
}
```

Qual a complexidade dos métodos acima?



Fila Encadeada – Enfileirar (enqueue)

```
class FilaEnc1 {  
    ...  
    void enfileirar(char v){  
        NoFila1* no = new NoFila1;  
        no->valor = v;  
        no->prox = inicio;  
        inicio = no;  
        N=N+1;  
    }  
}
```

Qual a complexidade do método acima?



Fila Encadeada – Frente (front)

```
class FilaEnc1 {  
    ...  
    char frente(){  
        NoFila1* p = inicio;  
        while(p->prox->prox != NULL) // dois na frente  
            p=p->prox;  
        char r = p->prox->valor;  
        delete p->prox;  
        p->prox = NULL;  
        N=N-1;  
        return r;  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** corrija o método para $N < 2$.



Fila Encadeada – Desenfileirar (dequeue)

```
class FilaEnc1 {  
    ...  
    char desenfileirar(){  
        NoFila1* p = inicio;  
        while(p->prox->prox != NULL) // dois na frente  
            p=p->prox;  
        char r = p->prox->valor;  
        delete p->prox;  
        p->prox = NULL;  
        N=N-1;  
        return r;  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** corrija o método para $N < 2$.



Fila Encadeada – Limitações

- ▶ A Fila Encadeada apresentada tem complexidade linear em algumas operações.
- ▶ Isto implica que se o número de elementos na fila for muito grande (ex. 1 bilhão), o tempo necessário será muito grande!
- ▶ **Tarefa:** implemente novos métodos em tempo $O(1)$! Dica: você pode alterar a classe FilaEnc1 para incluir novas variáveis, incluindo novos apontadores...
- ▶ **Desafio:** desenvolva uma fila encadeada com tempos $O(1)$ para as 3 operações, utilizando uma lista duplamente encadeada.



Filas Sequenciais

As Filas Sequenciais utilizam um array para armazenar as informações, então as informações sempre estarão em um espaço contíguo de memória.



Fila Sequencial – Definição

Consideraremos uma fila com elementos do tipo caractere.

```
#define MAXN = 100000
class FilaSeq1 {
    public:
        char elementos[MAXN];
        int N; // numero de elementos atualmente na fila

        void constroi() {...}
        void destroi() {...}

        char frente() {...}
        void enfileirar(char valor) {...}
        char desenfileirar() {...}
}
```

Qual a complexidade de espaço da fila sequencial?



Fila Sequencial – Construção e Destruição

A operação `constroi` inicializa a fila para uso, e a função `destrói` libera os recursos requisitados.

```
class FilaSeq1 {  
    ...  
  
    void constroi(){  
        N = 0;  
    }  
  
    void destrói(){  
        // vazio  
    }  
}
```

Qual a complexidade dos métodos acima?



Fila Sequencial – Frente

A operação de frente em uma fila sequencial retorna o primeiro elemento a entrar.

```
class FilaSeq1 {  
    ...  
    void frente(){  
        return elementos[0];  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** imprima uma mensagem de erro caso a fila esteja vazia.



Fila Sequencial – Enfileirar (enqueue)

A operação de enfileirar em uma fila sequencial adiciona um novo elemento no fim da fila.

```
class FilaSeq1 {  
    ...  
    void enfileirar(char valor){  
        elementos[N] = valor;  
        N=N+1;  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** imprima uma mensagem de erro caso a pilha esteja cheia.



Fila Sequencial – Desenfileirar (dequeue)

A operação de desenfileirar em uma fila sequencial remove e retorna o primeiro elemento da fila.

```
class FilaSeq1 {  
    ...  
    char desenfileirar(){  
        char r = elementos[0];  
        for(int i=0; i<N-1; ++i)  
            elementos[i] = elementos[i+1];  
        N=N-1;  
        return r;  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** imprima uma mensagem de erro caso a fila esteja vazia.



Fila Sequencial – Aprimoramentos

A Fila Sequencial proposta é bastante ineficiente ao desenfileirar. Será possível evitar a realocação dos elementos da fila para desenfileirar em tempo constante $O(1)$?



Fila Sequencial 2 – Definição

Consideraremos uma fila com elementos do tipo caractere.

```
#define MAXN = 100000
class FilaSeq2 {
    public:
        char elementos[MAXN];
        int N; // numero de elementos atualmente na fila
        int inicio;
        int fim;

        void constroi() {...}
        void destroi() {...}

        char frente() {...}
        void enfileirar(char valor) {...}
        char desenfileirar() {...}
}
```

Qual a complexidade de espaço da fila sequencial??



Fila Sequencial 2 – Construção e Destruição

A operação `constroi` inicializa a fila para uso, e a função `destroi` libera os recursos requisitados.

```
class FilaSeq2 {  
    ...  
  
    void constroi(){  
        N = 0;  
        inicio = 0;  
        fim = 0;  
    }  
  
    void destroi(){  
        // vazio  
    }  
}
```

Qual a complexidade dos métodos acima?



Fila Sequencial 2 – Frente

A operação de frente em uma fila sequencial retorna o primeiro elemento a entrar.

```
class FilaSeq2 {  
    ...  
    void frente(){  
        return elementos[inicio];  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** imprima uma mensagem de erro caso a fila esteja vazia.



Fila Sequencial 2 – Enfileirar (enqueue)

A operação de enfileirar em uma fila sequencial adiciona um novo elemento no fim da fila.

```
class FilaSeq1 {  
    ...  
    void enfileirar(char valor){  
        elementos[fim] = valor;  
        fim = fim + 1;  
        N=N+1;  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** imprima uma mensagem de erro caso a fila esteja cheia.



Fila Sequencial 2 – Desenfileirar (dequeue)

A operação de desenfileirar em uma fila sequencial remove e retorna o primeiro elemento da fila.

```
class FilaSeq2 {  
    ...  
    char desenfileirar(){  
        char r = elementos[inicio];  
        inicio = inicio + 1;  
        N=N-1;  
        return r;  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** imprima uma mensagem de erro caso a fila esteja vazia.



Fila Sequencial – Novos Aprimoramentos

A Fila Sequencial proposta tem tempo constante nas operações =)
Porém... ela é limitada em relação ao número de operações de
enfileirar e desenfileirar =(
Qual é este limite? Será possível corrigir esta falha?



Fila Sequencial 3 – Definição

Consideraremos uma fila com elementos do tipo caractere.

```
#define MAXN = 100000
class FilaSeq3 {
    public:
        char elementos[MAXN];
        int N; // numero de elementos atualmente na fila
        int inicio;
        int fim;

        void constroi(){...} // idem FilaSeq2
        void destroi(){...} // idem FilaSeq2

        char frente(){...} // idem FilaSeq2
        void enfileirar(char valor){...}
        char desenfileirar(){...}
}
```

Qual a complexidade de espaço da fila sequencial??



Fila Sequencial 3 – Enfileirar (enqueue)

A operação de enfileirar em uma fila sequencial adiciona um novo elemento no fim da fila.

```
class FilaSeq3 {  
    ...  
    void enfileirar(char valor){  
        elementos[fim] = valor;  
        fim = (fim + 1) % N;  
        N=N+1;  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** imprima uma mensagem de erro caso a fila esteja cheia.



Fila Sequencial 3 – Desenfileirar (dequeue)

A operação de desenfileirar em uma fila sequencial remove e retorna o primeiro elemento da fila.

```
class FilaSeq3 {  
    ...  
    char desenfileirar(){  
        char r = elementos[inicio];  
        inicio = (inicio + 1) % N;  
        N=N-1;  
        return r;  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** imprima uma mensagem de erro caso a fila esteja vazia.



Fila Sequencial – Limitações

Existe a limitação física de $MAXN$ posições imposta pela alocação estática.

Desafio: implemente uma Fila Sequencial utilizando **alocação dinâmica** para o vetor elementos. Assim, quando não houver espaço para empilhar novos elementos, crie mais espaço na memória (de preferência com o dobro do tamanho anterior) e mova o conteúdo atual da pilha para esse novo vetor. Dica: você vai precisar de alterar os métodos `constroi()`, `destroi()` e também inserir uma nova variável na Fila para permitir a variação do limite máximo de elementos.



Fila – Revisão

- ▶ Quais são os 3 métodos de uma fila?
- ▶ Qual é a complexidade de cada método em uma Fila Sequencial?
- ▶ Qual é a complexidade de cada método em uma Fila Encadeada?
- ▶ Quais as vantagens e desvantagens de cada implementação de fila?



Bibliografia

Szwarcfiter, J.L.; Markenzon, L. Estruturas de Dados e seus Algoritmos. Rio de Janeiro, LTC, 1994.

Bibliografia Adicional:

- ▶ Cerqueira, R.; Celes, W.; Rangel, J.L. Introdução a estruturas de dados: com técnicas de programação em C. Editora, 2004.
- ▶ Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein Algoritmos: Teoria e Prática. Ed. Campus, 2002.
- ▶ Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. Introduction to Algorithms, 3rd ed.. The MIT Press, 2009.
- ▶ Preiss, B.R. Estruturas de Dados e Algoritmos Ed. Campus, 2000;
- ▶ Knuth, D.E. The Art of Computer Programming - Vols I e III. 2nd Edition. Addison Wesley, 1973.
- ▶ Graham, R.L., Knuth, D.E., Patashnik, O. Matemática Concreta. Segunda Edição, Rio de Janeiro, LTC, 1995.