

Estruturas de Dados

Helena Graziottin Ribeiro
hgrib@ucs.br

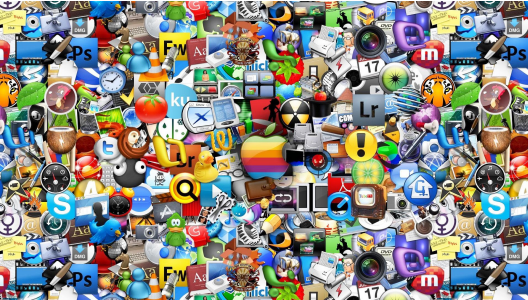
Programas de computador

São desenvolvidos para tratar problemas:

- armazenar informação
- recuperar informação



Como desenvolver programas, sistemas?

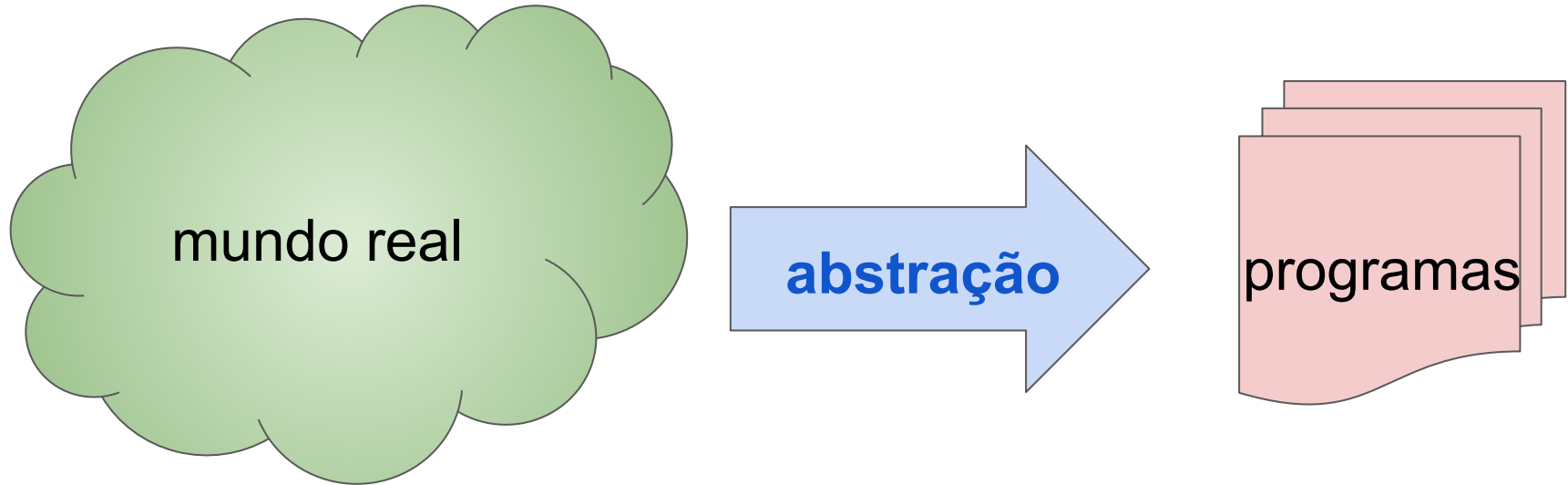


Como desenvolver programas, sistemas?

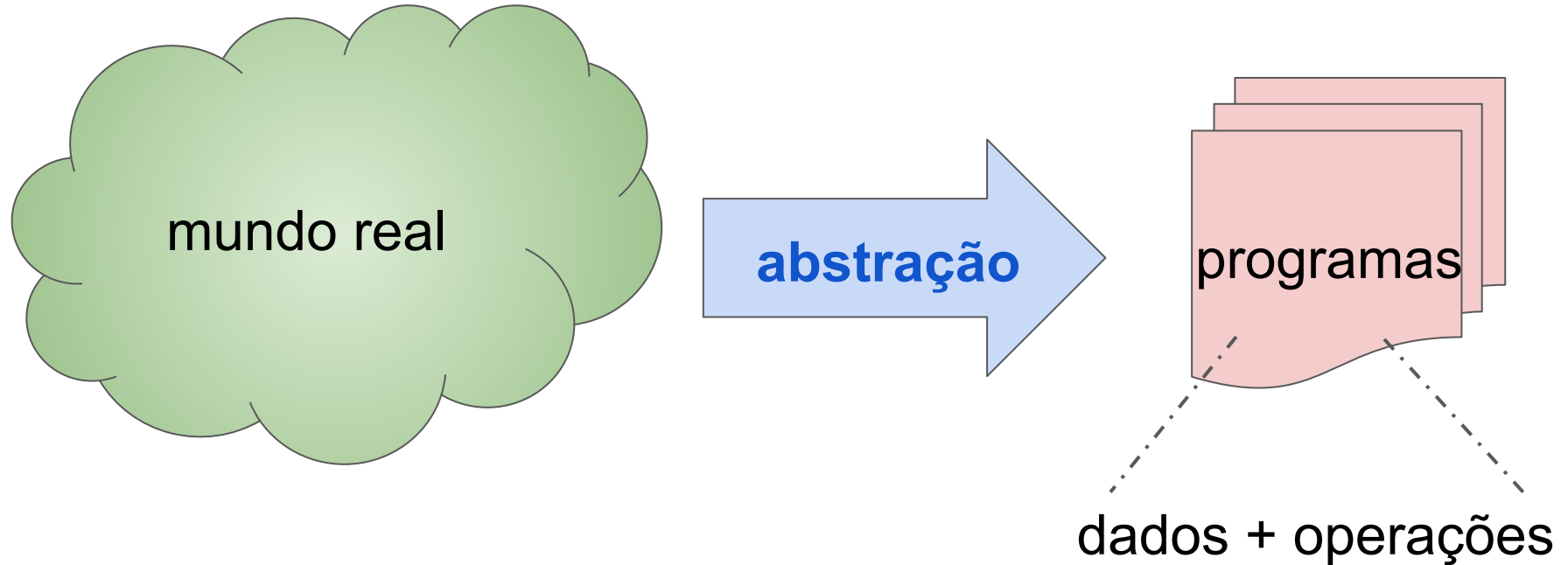
abstração:
análise,
requisitos,
recursos



Como desenvolver programas, sistemas?



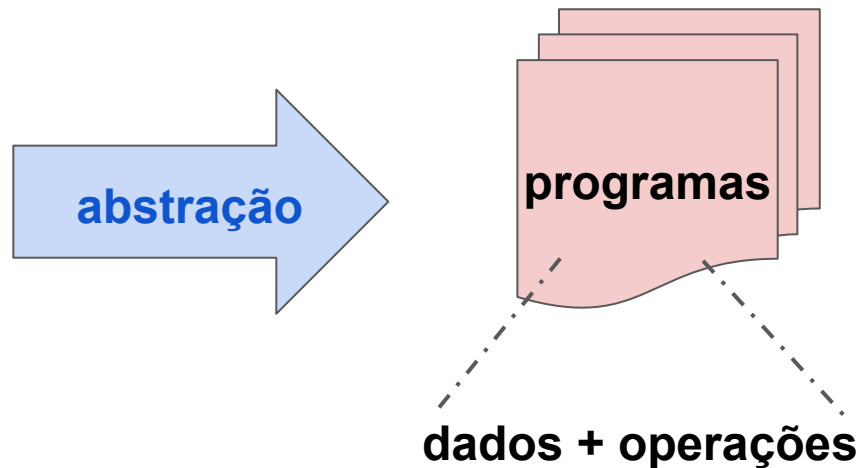
Como desenvolver programas, sistemas?



Como desenvolver programas, sistemas?

Na análise do problema, define-se:

Estruturas de Dados = Dados a Representar + Operações Associadas



Estruturas de Dados

- Define uma **organização** ou **estruturação** para um conjunto de dados
- **Independente da implementação** em uma linguagem de programação):
 - definições mais abstratas,
 - podem ser implementadas em diferentes linguagens

Estruturas de Dados

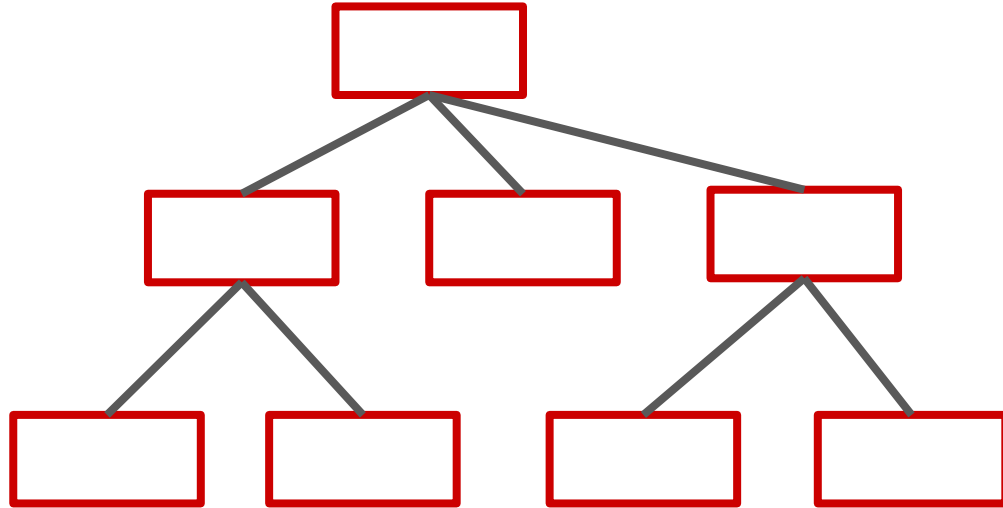
Assim:

- **Dados a representar:** número inteiro, positivo, negativo, número fracionário, nome, lista de nomes, etc..
- **Operações associadas:** inserção, busca, remoção, alteração, impressão, ordenação, etc...

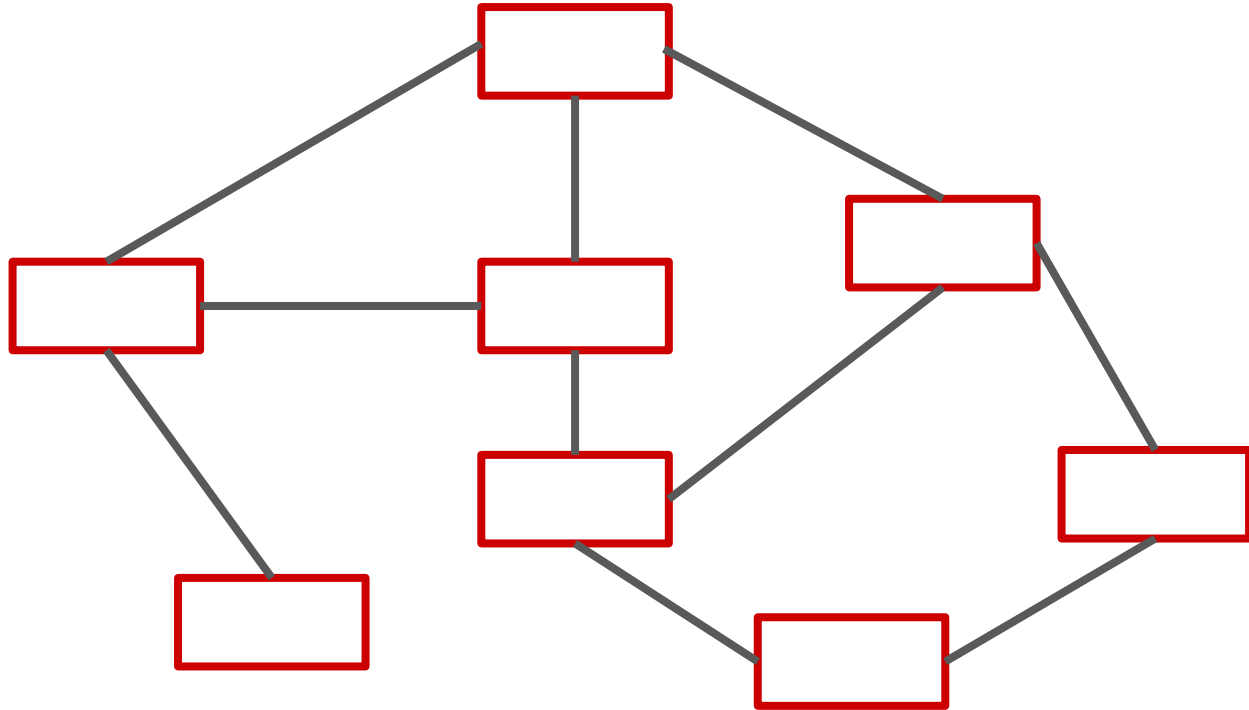
Estruturas de Dados



Estruturas de Dados

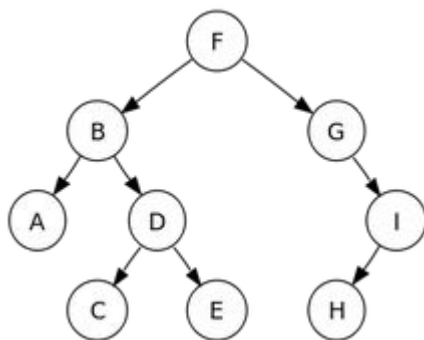
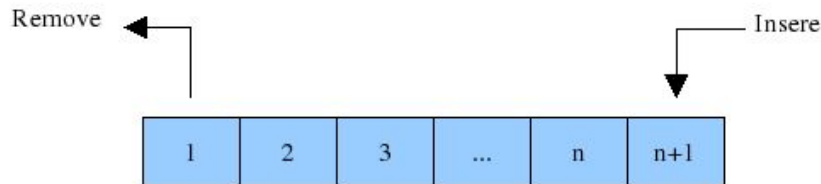


Estruturas de Dados



Estruturas de Dados clássicas

- listas
 - pilhas
 - filas
- árvores
- grafos



Índice	
1	
2	
3	
4	
5	

Chave	Registro
1	Maria
2	Ana
3	Paulo
4	Rodrigo
5	Carlos

Estruturas de Dados específicas

Exemplo: lista de nomes de pessoas

- **estrutura de dados:** lista
- **dados a representar:** nomes de pessoas
- **operações associadas:** inserir um nome em ordem alfabética, alterar um nome já presente na lista, remover um nome da lista, gerar uma sub-lista só com nomes que iniciam por uma determinada letra, imprimir um relatório com os nomes em ordem alfabética

Estruturas de Dados clássicas

As estruturas de dados são implementadas em um programa através de uma linguagem de programação, de acordo com as **estruturas de representação** (ou de implementação) que a linguagem oferece:

Estruturas de Dados = dados a representar + operações associadas



Estruturas de Representação = tipos de dados + funções, métodos

Estruturas de Dados X Estruturas de representação

abstração

Estruturas de Dados = dados a representar + operações associadas

Estruturas de Representação = tipos de dados + funções, métodos

Estruturas de Dados X Estruturas de representação

Estruturas de Dados = dados a representar + operações associadas

Estruturas de Representação = tipos de dados + funções, métodos

linguagens de programação

Estruturas de Dados X Estruturas de representação

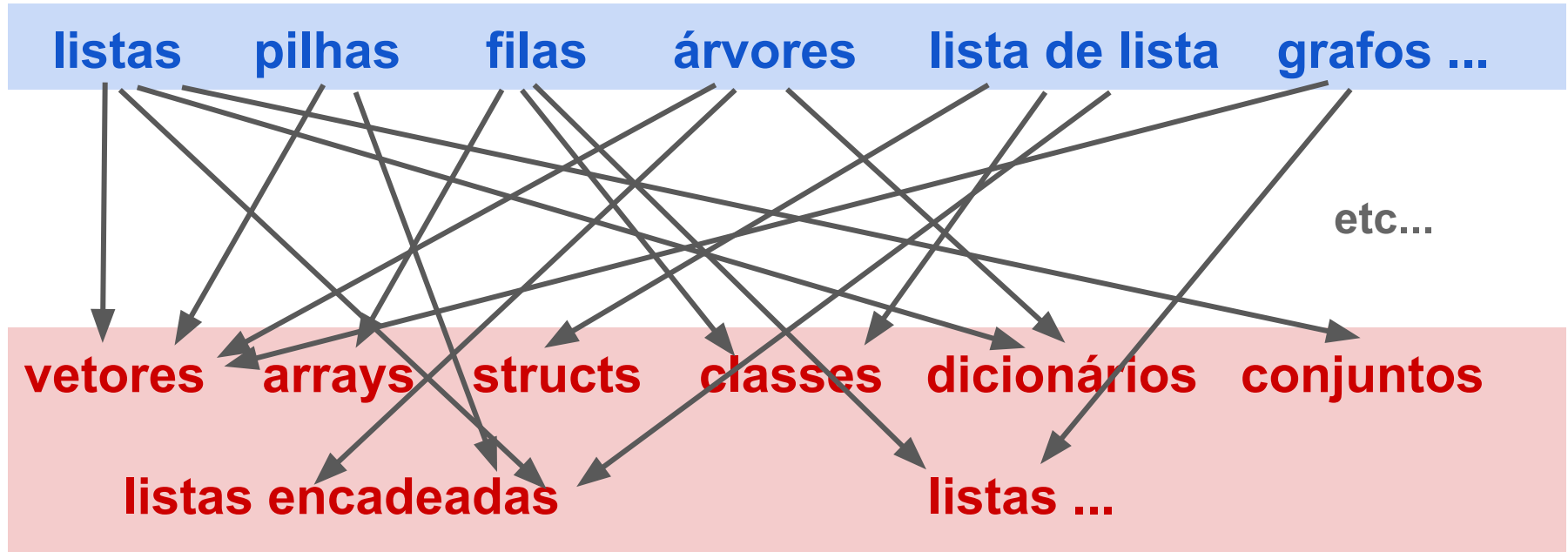
listas pilhas filas árvores lista de lista grafos ...



vetores arrays structs classes dicionários conjuntos

listas encadeadas simples listas encadeadas duplas...

Estruturas de Dados X Estruturas de representação



Desenvolvendo programas

Os programas de computador devem organizar a informação de forma a suportar o processamento **eficiente** (resolve o problema).

O que determina a **eficiência** de um programa?

- espaço de armazenamento
- tempo de processamento

Desenvolvendo programas

- Primeiro pensar em:

**dados
necessários**

**operações que devem
ser realizadas**

depois, na implementação, definir/escolher organização dos dados que permita realizar as operações

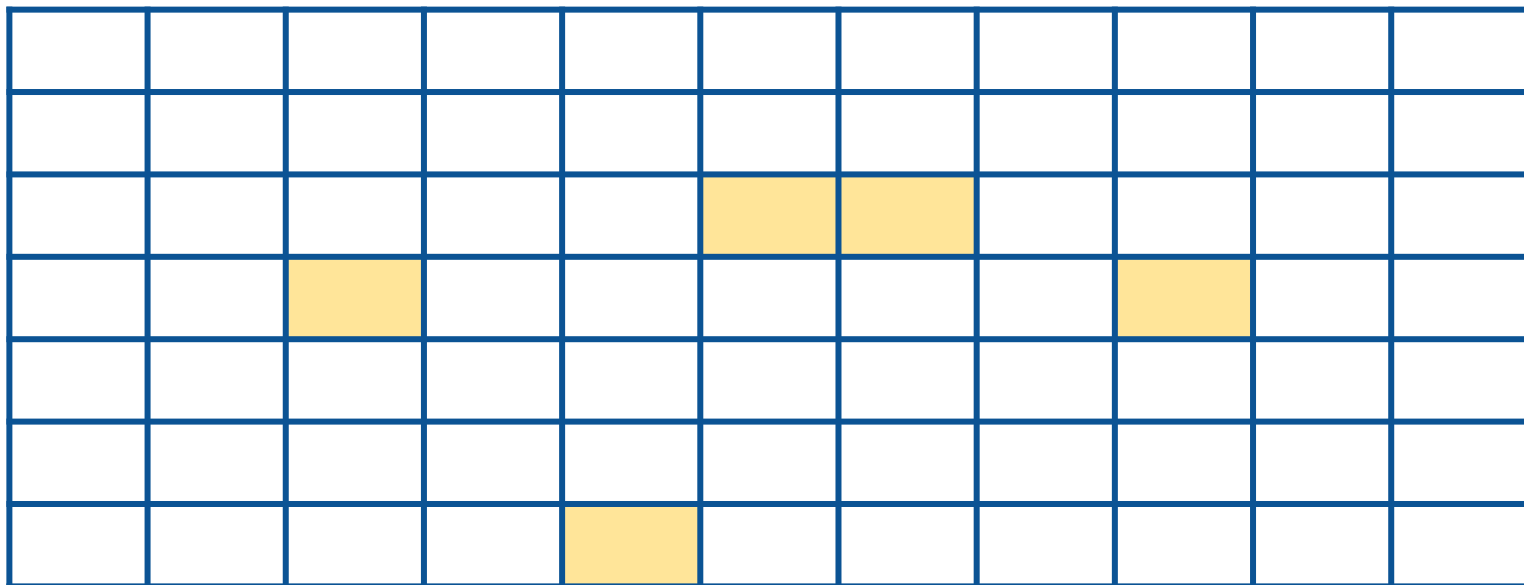
Desenvolvendo programas - armazenamento

Uso da memória principal nos programas:

- **memória estática:** alocação estática de variáveis globais e locais
- **memória dinâmica:** alocação dinâmica de memória
 - listas encadeadas

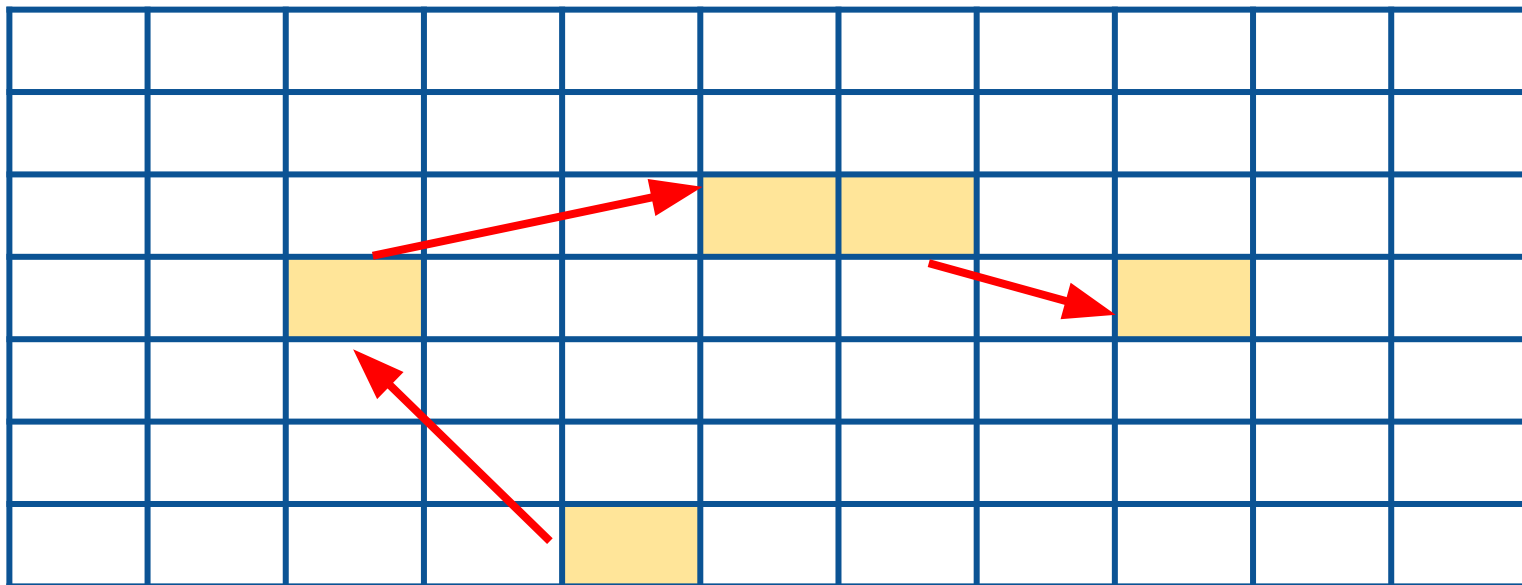
Desenvolvendo programas - armazenamento

Memória principal: **variáveis, alocação dinâmica**



Desenvolvendo programas - armazenamento

Memória principal: **alocação dinâmica**



Desenvolvendo programas

Em geral, as operações de inserção, remoção e busca definem a escolha de uma **estrutura de representação (organização dos dados) que permita execuções mais eficientes**. Algumas perguntas-chave:

- Todos os dados são inseridos no início ou inserções acontecem em vários momentos?
- Os itens de dados podem ser removidos?
- Todos os itens são processados em uma ordem definida, ou acesso aleatório é permitido?
- O conjunto de itens de dados é limitado ou pode crescer infinitamente?

Desenvolvendo programas

Tipicamente, deve-se analisar:

- se inserções podem ser misturadas com outras operações (por exemplo, ordenação)
- se as inserções/remoções são constantes ou esporádicas
- quais os tipos de consultas mais frequentes

Desenvolvendo programas

Custos e benefícios devem ser considerados:

- cada estrutura de representação terá custos e benefícios associados
- não há uma estrutura que seja melhor que outra em todas as situações
- estruturas de representação requerem:
 - **espaço** para cada item de dado,
 - **tempo** para cada operação básica,
 - **esforço de programação** (e manutenção)

TAD: Tipo Abstrato de Dado

- Definição genérica de uma estrutura de dados
- Um TAD define um tipo de dados somente em termos de um tipo e um conjunto de operações sobre aquele tipo.
- Um problema pode ser definido por mais de um TAD

```
TAD abc{  
  Dados:  
    ...  
  Operações:  
    ...  
}
```

TAD: exemplo

TAD conta {

Dados: nrconta: inteiro;

nomecliente: string;

saldo: real;

Operações:

saldo (E: nrconta; S: saldo

depósito (E: nrconta, valor; S: saldo);

saque (E: nrconta, valor; S: saldo);

...

}

TAD banco {

Dados: lista de contas,

nome do gerente: string

Operações:

cria_conta (E: nomecliente; S: nrconta);

somasaldos(S: valortotal);

saldospositivos(S: lista de nrcontas);

remove_conta(E: nrconta)

}