

Introdução

Prof. Marcelo de Souza

UDESC Ibirama
Bacharelado em Engenharia de Software

marcelo.desouza@udesc.br
Versão compilada em 13 de agosto de 2020

Leitura obrigatória:

- Capítulo 2 de [Edelweiss and Galante \[2009\]](#) – Conceitos básicos.

Leitura complementar:

- Capítulo 1 de [Pereira \[2008\]](#) – Introdução.

Conceitos básicos

Algumas definições:

- **Algoritmo:** sequência de passos para realizar uma tarefa.
- **Estrutura de dados:** forma sistemática de organizar e acessar os dados.

Objetivo: escolher os melhores componentes para resolver um problema.

Exemplo

Jogo para coleta de itens em uma grade (figura abaixo):

- ‘X’: item com valor 100.
- ‘O’: item com valor 200.
- Representação do cenário: matriz $n \times n$.
- Operações:
 1. Varrer ambiente.

2. Encontrar o item mais próximo.

									o				
			x										
								o			x		
		x											
							x						
		o										o	

Qual o consumo de memória e tempo de processamento?

- Cenário 10×10 .
- Cenário 1000×1000 .
- Cenário 10000×10000 .

Quanto tempo demora para executar as operações?
Quanta memória é consumida para armazenar o cenário?

Solução: usar uma estrutura de dados mais eficiente!

Linguagens de programação fornecem:

- Tipos de dados primitivos: inteiro, real, lógico...
- Tipos de dados estruturados: arranjos, registros, sequências...

Usar esses recursos para criar:

- TAD (tipo abstrato de dados): estruturas definidas pelo usuário.
 - Organização dos dados.
 - Operações sobre os dados.

Representação física:

- Contiguidade física
 - Ex: vetores e matrizes.
 - Valores armazenados sequencialmente na memória.
 - \oplus rápido acesso.
 - \ominus espaço físico estático.
- Encadeamento
 - Ex: listas, pilhas e filas.
 - Alocação (não sequencial) dinâmica de memória.
 - \oplus maleabilidade.
 - \ominus acesso serial.

Atividades

1. Implemente o exemplo do jogo para coleta de itens. Verifique a memória utilizada e o tempo de processamento das operações em função dos diferentes tamanhos da grade. Verifique qual o limite de tamanho capaz de ser processado.
2. Proponha uma nova estrutura de dados, capaz de utilizar menos memória e melhorar o desempenho na execução das operações. Verifique o novo limite de tamanho capaz de ser processado.

Referências

- Edelweiss, N. and Galante, R. (2009). *Estruturas de Dados: Volume 18*. Bookman Editora.
- Pereira, S. d. L. (2008). *Estruturas de dados fundamentais: Conceitos e aplicações*.