

Arrays e Vetores

Prof. Rafael Guterres Jeffman
rafael.jeffman@gmail.com

Arrays

- São as estruturas de dados mais simples que existem.
- Consistem em uma coleção de elementos identificados por um índice.
- Um *array* é armazenado de forma que a posição de cada elemento pode ser calculada a partir de um índice.
- Normalmente, os dados em um *array* são organizados de forma contígua na memória.

Aplicações de um Array

- *Arrays* são as estruturas de dados mais básicas existentes.
- A memória de um computador é organizada em forma de *array*.
- Qualquer estrutura de dado pode ser implementada utilizando-se um *array* como a base para o armazenamento dos dados.

Operações em um Array

- Leitura de um elemento: executada em $O(1)$.
- Escrita de um elemento: executada em $O(1)$.
- Procura de um elemento: executada em $O(n)$.
- *Arrays* não possuem as operações de inserção e remoção de elementos.
- O tamanho de um *array* é sempre igual a sua capacidade de armazenamento.

Arrays em Java

- Arrays em Java são dinâmicos, no sentido que seu tamanho é definido não pelo código, mas no momento de sua criação.
- Uma vez criados, um array em Java não pode ter seu tamanho alterado.
- Após a criação de um array de Objetos, o mesmo não contém nenhum objeto, e o seu tamanho reflete a capacidade de armazenamento de objetos.

Código Java

```
int[] intArray = new int[100];  
String[] stringArray = new String[100];
```

- intArray contém 100 “inteiros”, cujo valor é ZERO.
- stringArray contém 100 referências à objetos String, onde todos são **null**.
- Um array pode conter “buracos”, ou seja, nem todo elemento em um array contém um valor válido e/ou definido.

Vetores

- Vetores são estruturas de dados semelhantes a *arrays*, com a diferença que não possuem “buracos”, e possuem as operações de inserção e exclusão.
- Esta definição de vetores não é comum na bibliografia existente. Utilizaremos esse conceito para diferenciar do comportamento de *Arrays*.

Operações em um Vetor

- Inclusão e Exclusão
 - Como um vetor não possui “buracos”, estas operações fazem com que os dados armazenados tenham que ser “deslocados” dentro do vetor.
 - Este deslocamento aumenta o número de operações de movimentação de dados em memória, diminuindo a eficiência da estrutura.
 - Um vetor é bastante eficiente para inclusão e remoção de elementos na última posição válida.

Características de um Vetor

- Normalmente, um vetor não tem um tamanho pré-definido, sendo que seu tamanho é alterado sempre que necessário.
- Uma operação comum em um Vetor é a adição de um elemento no fim da estrutura (*append*).

Implementação em Java

- Os dados de um vetor são normalmente armazenados em um *array*.
- Uma vez que o *array* fique “cheio”, o seu tamanho deve ser incrementado, e como isso é impossível em Java, um novo *array* deve ser criado e os dados do original copiados para este novo *array*.
- Esta operação faz com que seja consumido o dobro da memória necessária para um *array*, quando ocorre uma operação de inserção.

Exercícios

- Implemente, em Java, um Vetor com as características estudadas.
- Qual o maior problema de *velocidade de execução* no código implementado e como melhorar esse resultado?
- Como implementar um Vetor genérico para qualquer tipo de classe? Você consegue fazer isso sem *warnings* no código?