

Técnicas de Programação

Tipos abstratos de dados

Algoritmo: sequência de passos (finita) para resolver um problema

Algoritmos

- é garantido que acaba
- dá a resposta correta 100% das vezes
- cada passo é bem definido
- dada uma entrada, devolve sempre a mesma resposta
- independe da linguagem de programação usada

Algoritmos

Precisamos armazenar dados. Até agora sabemos usar os seguintes recursos em Java e Python

- listas, dicionários para dados "puros"
- classes para dados + comportamentos

Problemas

Adicionar no fim da lista: devo usar `list.append` ou `ArrayList.add` ?

1. não conhecemos como list/dict/ArrayList/TreeMap funcionam
2. APIs diferentes dificultam saber exatamente o que queremos dizer

`list.append` ou `ArrayList.add` não são passos **bem definidos**

Tipo Abstrato de Dados

Nomear uma estrutura para armazenamento de dados e definir

- conjunto de operações válidas
- restrições na lógica dessas operações
- *complexidade computacional* ("custo") de cada operação

Implementação pode ocorrer em qualquer linguagem

Array

Tipo de dados mais primitivo e representa uma coleção de objetos contínua indexada por um número inteiro começando em `0`. Tem tamanho fixo.

- `NOVO_ARRAY(N)` - cria array com capacidade `N > 0`
- `TAMANHO(A)` - devolve o número de elementos do array
- `A[i]` - devolve o elemento de índice `i`. Se `i < 0` ou `i > TAMANHO(A)` dá erro

Exemplo

- **Entrada:** Array `A`, inteiro `v`
- **Saída:**
 - `Verdadeiro` se o array contém o elemento `v`
 - `Falso` caso contrário

```
PARA CADA i = 0 ATÉ TAMANHO(A) FAÇA
    SE A[i] = valor ENTÃO
        RETORNE Verdadeiro
    FIM
FIM

RETORNE Falso
```


E para implementar?

- Em Python, o tipo `list` pode ser usado para representar um array
- Em Java temos o tipo `Array` "de verdade", mas poderíamos usar um `ArrayList` também

Todas essas opções respeitam os comportamentos definidos na ADT `Array`

Módulo ADT

- Arrays e listas: aprendemos a usar array e implementamos o tipo `List`
- Listas e dicionários: implementamos o tipo `Map` com listas
- Complexidade computacional: comparar diferentes implementações da mesma ADT.