

Cap. 1.2 - Técnicas de resolução de problemas

Aprenda a Programar com C# - 2ª Edição (2020)

Edições Sílabo

<https://bit.ly/36nyKFm>

António Trigo, Jorge Henriques
{antonio.trigo,jmvhenriques}@gmail.com

1 de outubro de 2020

Abordagem

Noção de algoritmo

Formas de representação de algoritmos

Descritiva

Pseudocódigo

Fluxograma

Abordagem

- ▶ Na construção de uma solução para um problema há que identificar 3 partes:
 - ▶ Entradas, dados do problema;
 - ▶ Saídas ou resultados, que é o que se pretende obter;
 - ▶ Transformação, que são os passos necessários para converter as entradas em saídas (algoritmo do problema)

Análise de um problema

1. Perante o problema, analisá-lo e dividi-lo nas suas três partes: inicial, final e transformação (algoritmo);
2. Antes de se passar à obtenção de uma solução, é necessário compreender claramente os estados inicial e final;
3. Nesta fase, obtém-se uma visão geral do problema e da transformação necessária à sua resolução (não se procura ainda uma resolução detalhada).

Detalhe do algoritmo

1. Decompor a descrição geral do algoritmo num conjunto, conciso e ordenado, de passos que permitem transformar o estado inicial no estado final (i.e., dividir em passos mais específicos);
2. Examinar passos definidos em 1. e, para cada um destes, verificar se ainda podem ser divididos. Em caso afirmativo, proceder à sua divisão;
3. Repetir o processo em 2. até que todos os passos sejam não ambíguos (ser entendíveis por quem os vai executar).

Validação do algoritmo

1. Percorrer e executar todos os passos do algoritmo, na tentativa de identificar possíveis incorreções e assim garantir a correção dos resultados obtidos;
2. Deve-se ter em conta todos os casos válidos (possíveis estados iniciais ou dados de entrada);
3. Deve-se dar particular atenção aos casos extremos do conjunto de dados possíveis.

Exemplo: confeção de uma kiche

- ▶ Análise do problema
 - ▶ Estado inicial (entradas), são os ingredientes: 4 ovos, 1 pacote de natas, meia cebola, salsichas, cogumelos, fiambre, legumes, ...
 - ▶ Estado final (saídas), kiche
 - ▶ Transformação: misturar e cozinhar os ingredientes de forma a produzir a kiche.
- ▶ Detalhe do algoritmo:
 - ▶ misturar os ingredientes;
 - ▶ cozinhar a kiche.
- ▶ Este nível de detalhe ainda não é suficiente para a maioria das pessoas!

Exemplo: confeção de uma kiche

- ▶ Detalhe do algoritmo:
 - ▶ Misturar os ingredientes:
 - ▶ Juntar a cebola, os ovos e as natas;
 - ▶ Triturar;
 - ▶ Cozinhar a kiche:
 - ▶ Colocar o forno à temperatura desejada (200°C)
 - ▶ Colocar a massa na forma e colocar os ingredientes (salsichas, cogumelos, fiambre, legumes, ...);
 - ▶ Adicionar a mistura preparada no passo anterior;
 - ▶ Colocar a forma no forno e esperar o tempo de cozedura (30 minutos sem turbo).
- ▶ Este nível de detalhe poderá ainda ser insuficiente para utilizadores inexperientes (instruções de como utilizar o forno);
- ▶ Validação: comer a kiche.

Outro exemplo: algoritmo para fritar um ovo

1. Retirar o ovo do frigorífico;
2. Colocar a frigideira aquecer;
3. Colocar o óleo;
4. Esperar até o óleo ficar quente;
5. Partir o ovo separando a casca;
6. Colocar o conteúdo ovo na frigideira;
7. Esperar um minuto;
8. Retirar o ovo da frigideira;
9. Fechar o bico de gás;

Então, o que é um algoritmo?

- ▶ Sequência de instruções finitas e ordenadas de forma lógica para a resolução de uma determinada tarefa ou problema;
- ▶ Não necessariamente envolve aspetos computacionais. Ex.: Uma receita de bolo, trocar um pneu de carro, trocar uma lâmpada, manual de instruções;
- ▶ O algoritmo não é a solução do problema, mas o caminho que leva a mesma!

Um algoritmo deve

- ▶ Descrever todas as transformações necessárias para resolver o problema (de forma executável pelo computador);
- ▶ Ser não ambíguo, de forma a que pessoas distintas que o sigam (corretamente) obtenham os mesmos resultados;
- ▶ Para o mesmo conjunto de dados os resultados deverão ser sempre os mesmos;
- ▶ Conduzir a resultados corretos para qualquer conjunto de dados válido;
- ▶ todos os passos de um algoritmo devem ser explicitados utilizando unicamente operações que o computador seja capaz de executar: receber e fornecer informação; guardar informação para posterior utilização; operações aritméticas; operações lógicas; escolher entre várias ações – seleção; repetir um conjunto de ações – repetição.

Descritiva - exemplo

- ▶ Soma dos números de uma pilha de cartões:
 1. Ver e anotar o 1^o número da pilha de cartões;
 2. Retirar o cartão que está no topo;
 3. Se a pilha estiver vazia (já vimos todos os cartões) então passar ao passo 6;
 4. Senão, ver o número que está no topo da pilha e somá-lo com o anotado. A soma passa a ser o novo número anotado;
 5. Recomeçar no passo 2;
 6. Escreve o resultado – soma dos números na pilha de cartões.

Descritiva - exemplo

- ▶ Soma dos números de uma pilha de cartões, do ponto de vista do computador:
 1. Iniciar a zero um valor (soma);
 2. Ler o primeiro número (valor);
 3. Repetir as instruções seguintes enquanto valor ≥ 0 ;
 - 3.1 Somar valor ao que já estiver em soma;
 - 3.2 Ler o próximo valor;
 4. Escrever o resultado (soma).

Pseudocódigo

- ▶ É uma forma genérica de escrever um algoritmo, utilizando uma linguagem simples (nativa a quem o escreve, de forma a ser entendida por qualquer pessoa) sem necessidade de conhecer a sintaxe de nenhuma linguagem de programação;
- ▶ É, como o nome indica, um pseudocódigo (código falso) e, portanto, não pode ser executado num sistema real (computador);
- ▶ Permite ao programador expressar as suas ideias sem ter que se preocupar com os detalhes sintáticos de uma linguagem de programação;
- ▶ Forma de escrita de um algoritmo que permite, facilmente, a sua tradução para uma linguagem de programação;
- ▶ É muito utilizado nos livros para ilustrar algoritmos de forma a que todos os programadores entendam (independentemente da linguagem que utilizem).

Pseudocódigo

- ▶ A escrita do pseudocódigo inicia-se sempre com a palavra-chave **início** e termina sempre com a palavra-chave **fim**;
- ▶ As palavras-chave do pseudocódigo não podem ser utilizadas em expressões, constantes ou variáveis;
- ▶ As palavras-chave deverão aparecer sempre destacadas ou em maiúsculas;
- ▶ Aconselha-se o recurso à indentação, sempre que se refere a instruções dentro de uma estrutura (a leitura torna-se mais fácil);
- ▶ Para a introdução de linhas de comentário recorre-se à utilização do asterisco (*), no início da linha.

Pseudocódigo - Exemplo 1

Algorithm 1 Soma dos números de um conjunto

inicio

soma = 0

ler *valor*

enquanto *valor* ≥ 0 **fazer**

soma \leftarrow *soma* + *valor*

ler valor

fim enquanto

escrever soma

fim

Pseudocódigo - Exemplo 2

Algorithm 2 Verificar se número é par ou ímpar

início

ler *numero*

se $\text{numero} \% 2 = 0$ **então**

escrever *numero* e' par

senão

escrever *numero* e' ímpar

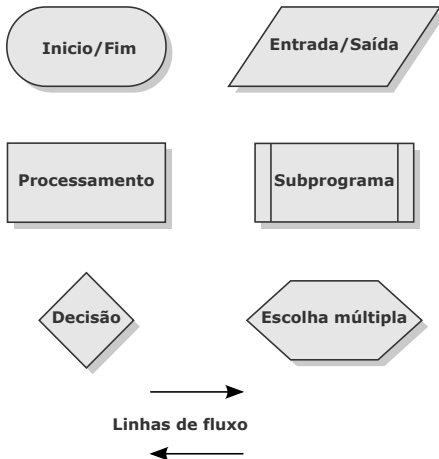
fim se

fim

Fluxogramas

- ▶ Diagramas representativos do fluxo de ações de um algoritmos através de símbolos;
- ▶ Permitem visualizar rapidamente a estrutura de um algoritmo;
- ▶ Cada tipo de instrução é representado por um símbolo específico;

Fluxogramas - símbolos



Fluxogramas - símbolos - exemplo

