

Os algoritmos estão em toda parte

Marisa Ortegoza da Cunha
marisa.ortegoza@bol.com.br

Uma crítica comum que se faz ao ensino e uso de algoritmos é que ele se contrapõe aos processos criativos, à atribuição de significado às ações realizadas. A repetição linear e automática representaria um obstáculo ao novo, ao original, à criação.

Algumas questões:

- O que é um algoritmo?
- Está presente somente no mundo da técnica, da Matemática?
- Podemos prescindir dele, totalmente?
- Necessitamos dele para tudo?
- Tudo é passível de algoritmização?
- Se não, como agir diante de uma tarefa não algoritmizável?
- O procedimento algorítmico se opõe à criatividade?

Etimologia :

Há, basicamente, duas versões para a origem do termo algoritmo:

- seria derivado do nome de um matemático persa chamado Abu Ja'far Maomé ibn Mûsâ al-Khowârizm (825) que, traduzido literalmente, quer dizer Pai de Ja'far, Maomé, filho de Moisés, de Khowârizm. Khowârizm é hoje a cidade de Khiva, na ex União Soviética. Escreveu, entre outras obras, "Algorithmi de numero indorum", sobre os algoritmos usando o sistema de numeração decimal (indiano). Escreveu também um livro chamado *Kitab al jabr w' al-muqabala* (Regras de Restauração e Redução), cujo título deu origem à palavra Álgebra. Mesma origem da palavra *algarismo*.

- seria derivado da palavra *Al-goreten*, que significa raiz, conceito que se pode aplicar aos cálculos.

A possibilidade de que a palavra algoritmo derive de uma que contenha a idéia de número pode nos levar a considerar que a definição de uma seqüência, que é a marca dos algoritmos, esteja relacionada com a seqüência dos números ordinais. Há sempre um primeiro passo, seguido de um segundo, e assim por diante, na execução de qualquer processo algorítmico. Assim, a característica mais marcante de um algoritmo é se constituir numa **seqüência** de passos simples, para a realização de uma tarefa mais complexa.

Para Descartes, essa era a estratégia primordial para a resolução de problemas mais complexos - a marca dos algoritmos é, então, o encadeamento cartesiano, a linearidade, a complexidade fragmentada na simplicidade.

Os algoritmos não são produto da era do cálculo automatizado - mesmo considerando-se seu caráter de realizador de cálculos (como destacado no dicionário Aurélio: um algoritmo é um "Processo de cálculo, ou de resolução de um grupo de problemas semelhantes, em que se estipulam, com generalidade e sem restrições, regras formais para a obtenção de resultado ou de solução de problema."), temos conhecimento de algoritmos que datam de séculos antes de Cristo, e que a maioria de nós estudou no ensino fundamental.

Os exemplos mais conhecidos são:

- o algoritmo de Euclides (um dos mais antigos - apareceu na obra *Elementos*, por volta de 300 a.C.), para a determinação do maior divisor comum de dois números inteiros positivos:

Algoritmo de Euclides

1. Dividir o maior número pelo menor;
2. Se o resto for zero, o número menor será o mdc procurado;
3. Se o resto não for zero, fazer o número menor tomar o lugar do maior e o resto tomar o lugar do menor;
4. Voltar para o passo 1.

- o *crivo de Eratóstenes* (devido a Eratóstenes de Cirene, séc. III a.C.), para identificar os números primos positivos menores do que um inteiro arbitrário:

Crivo de Eratóstenes

1. Escrever todos os inteiros de 1 a N;
2. Cortar, com um traço, todos os múltiplos de 2;
3. A cada passo seguinte, cortar todos os múltiplos do primeiro número ainda não cortado;
4. Repetir o passo 3 até que o próximo número não cortado seja igual ou superior à raiz quadrada de N. (Critério de suficiência)

Por outro lado, um algoritmo não representa, necessariamente, um programa de computador, e sim os passos necessários para realizar *qualquer* tarefa que admita uma subdivisão de complexidade hierarquicamente decrescente. Atualmente, é quase inevitável associarmos o termo algoritmo aos processos computacionais, mas tarefas descritas segundo regras simples encadeadas fazem parte do nosso dia-a-dia, como as receitas de bolo, a troca de um pneu, os gestos feitos ao nos vestirmos, entre inúmeros outros.

No âmbito da computação, adiciona-se uma condição aos processos algorítmicos: a *finitude*. Mesmo no cerne da Matemática não há um consenso a respeito de se aceitar ou não processos infinitos. Na segunda metade do século XIX, os matemáticos se dividiam em duas grandes escolas: de um lado, os finitistas - ou intuicionistas - que só aceitavam demonstrações ou processos finitos; de outro, os infinitistas, que incluíam tanto os formalistas como os logicistas, e que aceitavam, sim, aqueles procedimentos. De um lado, apenas seqüências eram aceitas; de outro, as séries (somadas infinitas) eram também objetos matemáticos.

As frações contínuas constituem um exemplo interessante de procedimento que é finito - quando operado sobre números racionais, e infinito - quando o número dado é irracional. A origem das frações contínuas está na Grécia, onde as frações, para efeito de comparações, eram todas escritas com numerador "1". A idéia do processo é, então, escrever um dado número real na forma

$$a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4 + \frac{1}{a_5 + \dots}}}}$$

Onde $a_1, a_2, \dots, a_n, \dots$ são inteiros positivos.

Por exemplo, $\frac{20}{7} = 2 + \frac{6}{7} = 2 + \frac{1}{\frac{7}{6}} = 2 + \frac{1}{1 + \frac{1}{6}}$ (processo finito)

Já a expansão de $\sqrt{2}$ em frações contínuas é infinita:

$$\sqrt{2} = 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \dots}}}$$

Embora o infinitismo tenha prevalecido no desenvolvimento da Matemática, é forçoso mencionar que Poincaré (falecido em 1912) foi, não só o grande representante do pensamento intuicionista, mas considerado o último grande matemático universalista, com conhecimentos amplos e profundos em diferentes subáreas da Matemática, como geometria e álgebra. Diz-se de Poincaré que, se Einstein não tivesse apresentado a teoria da relatividade, em 1905, o matemático francês o teria feito.

A questão finito x infinito se resolve pelas próprias limitações impostas pela tecnologia: toda a fundamentação matemática da computação deriva da escola finitista.

Um algoritmo absolutiza a explicitação. Mais do que resolver um problema, exige a descrição, passo a passo, do que deve ser feito. Claro que, no dia-a-dia, para a realização de atividades cotidianas (vestir-se, banhar-se, alimentar-se etc.), as pessoas se utilizam de algoritmos já internalizados e, portanto, o fazem de maneira intuitiva, sem que haja a conscientização de cada etapa do planejamento. Mas é justamente a existência de uma estrutura algorítmica subjacente a nossos gestos que nos possibilitam realizar essas seqüências "automáticas" de ações.

O processo de explicitar envolve, obrigatoriamente, uma linguagem, o que remete à quantidade de informação transmitida nessa explicitação. Cada etapa de um algoritmo contém ou transmite apenas 1 bit de informação, pois todas as decisões são tomadas a partir de respostas simples, do tipo sim/não, verdadeiro/falso. Improvisações ou ações baseadas em opinião, intuição, gosto, assim como os *insights*, tão presentes nas decisões das pessoas, não são minimamente contempladas por um esquema algorítmico.

O fato de que cada passo de um algoritmo se baseia numa decisão do nível mais simples possível é revelado pela própria caracterização de algoritmo no âmbito da computação: *um algoritmo seria aquilo - e somente aquilo - passível de ser executado por uma máquina de Turing.*

Máquina de Turing

Trata-se de uma máquina abstrata - uma experiência de pensamento - e pode ser imaginada como um aparelho mecânico pelo qual passa uma fita infinitamente longa com uma seqüência de casas que, ou estão vazias, ou contêm uma marca. A máquina pode executar apenas as seguintes operações ao ler uma casa - mover-se para a direita, mover-se para a esquerda, apagar uma marca, imprimir uma marca.

A máquina de Turing pode executar qualquer programa que possa ser expresso em código binário (isto é, traduzido em ações tomadas por decisões do tipo sim/não). O computador digital mais complexo é formalmente equivalente a uma máquina de Turing; a diferença (mais que relevante) reside nos tempos requeridos para a execução da tarefa, por um e por outro artefato.

No livro *O trabalho das nações*, Robert Reish - que foi secretário do trabalho do governo Carter - defende a existência de apenas duas classes de trabalhadores: os analistas e os seguidores de rotina. Os primeiros deteriam a visão, a teoria, e a eles caberia a tarefa intelectual de avaliar os problemas e traçar estratégias para sua resolução; no nosso contexto, a de *construir algoritmos*. Aos seguidores de rotina caberia exclusivamente a execução de tais algoritmos. Os analistas dariam conta da complexidade enquanto que os seguidores de rotina se limitariam a agir linearmente.

Pensemos na classificação proposta por Reish: será que existe algum setor das nossas vidas no qual possamos, realmente, agir *exclusivamente*, como analistas? Será que podemos prescindir totalmente de rotinas, de executar procedimentos encadeados, algorítmicos? O analista não será, em algum momento, *obrigatoriamente*, um seguidor de rotinas (quer sejam estabelecidas por ele próprio, quer sejam determinadas por terceiros)?

É claro, porém, que, se não podemos prescindir totalmente dos algoritmos, eles não dão conta de executar quaisquer tarefas ou resolver todos os problemas do nosso dia-a-dia. Como mencionado anteriormente, decisões que envolvam nossa intuição, nossa emoção, ações que realizamos movidos por razões inconscientes, fazendo uso de conhecimento que detemos mas que não somos capazes de explicitar, não podem ser traduzidas na forma simples e linear de um algoritmo.

Se tivermos que explicar "como" resolvemos um problema com essas características, ou seja, não-algoritmizável - provavelmente descreveremos um problema *parecido* com o original, este sim, de caráter algorítmico. Também assim

agem os matemáticos ou profissionais da computação para resolver um problema não linear: resolvem um problema linear, *próximo* (para algum sentido de *proximidade*) do problema inicial. Em outras palavras: por mais complexo que seja o problema enfrentado, se tivermos que comunicar, descrever ou explicar sua resolução a alguém, teremos, obrigatoriamente, que traduzi-lo (com maior ou menor grau de fidelidade) linearmente, de forma algorítmica.

Um computador representa a maximização do algoritmizável, mas ele viabiliza o uso de ferramentas que nos possibilitam alterar, corrigir, experimentar, com um grau de liberdade e segurança que antes não seria possível. Usado devidamente como meio, podemos, mais do que nunca, desenvolver e praticar nossa criatividade.¹

Para finalizar, para que este texto não se caracterize como um elogio radical à simplicidade, - um depoimento de quem leciona para estudantes de Ciência da Computação: **NADA** é tão difícil de ser ensinado como a elaboração de algoritmos.

E sabem por quê?

Porque a construção de algoritmos não é algoritmizável!!

Ou seja: estabelecer os passos *simples* que, encadeados, resolvem uma tarefa complexa é, em si, *uma tarefa complexa*!!

Bibliografia

- DAVID, Aurel. *A cibernética e o humano*. São Paulo: Hemus, 1971.
- MACHADO, Nilson J. - Os algoritmos devem ser ensinados? *In Pátio - Revista pedagógica*, fev/abr/2007 - n. 41, pp 48-51. Artmed Editora.
- PAGELS, Heinz R. *Os sonhos da razão - o computador e a emergência das ciências da complexidade*. Lisboa: Gradiva, 1990.
- PUGA, Sandra e RISSETTI, Gerson. *Lógica de programação e estrutura de dados*. São Paulo: Pearson - Prentice Hall, 2004.
- RUCKER, Rudy. *Mind tools - the mathematics of information*. London: Penguin Books, 1988.
- TEIXEIRA, João F. *Mentes e máquinas - uma introdução à ciência cognitiva*. Porto Alegre: Artes Médicas, 1998.
- WEIZENBAUM, Joseph. *O poder do computador e a razão humana*. Lisboa: Edições 70, 1976.

São Paulo, 10/06/2008

¹ MACHADO (2007) apresenta um artigo no qual discute a validade do ensino de algoritmos.