

Aula 1: Introdução

3/3/2015

Algoritmos

O que é um algoritmo?

Conjunto finito de passos para resolver um determinado problema computacional.

O que é um problema computacional?

Entrada: seqüência de inteiros a_1, \dots, a_n ;

Saída: os mesmos inteiros colocados em ordem crescente $b_1 < \dots < b_n$

Exemplos

- Soma, subtração, multiplicação, divisão
- Solução equação de segundo grau
- Construção de polígonos com régua e compasso
- Algoritmo de Euclides

Algoritmo de Euclides

Entrada: inteiros positivos a e b

Saída: o maior divisor comum de a e b

Enquanto o menor dos números não dividir o maior, subtrair o menor do maior. Quando o menor dividir o maior, então o menor é o maior divisor comum.

Proposição 2, Livro VII, dos Elementos



fl. 300 AC

Exercício

Descreva de forma clara, em português, como somar dois números representados na base 2:

- De modo que uma pessoa possa ler sua descrição e aprender a somar dois números.
- De modo que uma pessoa possa seguir os passos de sua descrição e obter ao final a soma dos dois números, sem necessariamente ter de compreender como fazer uma soma.

Computadores

Máquinas capazes de executar algoritmos



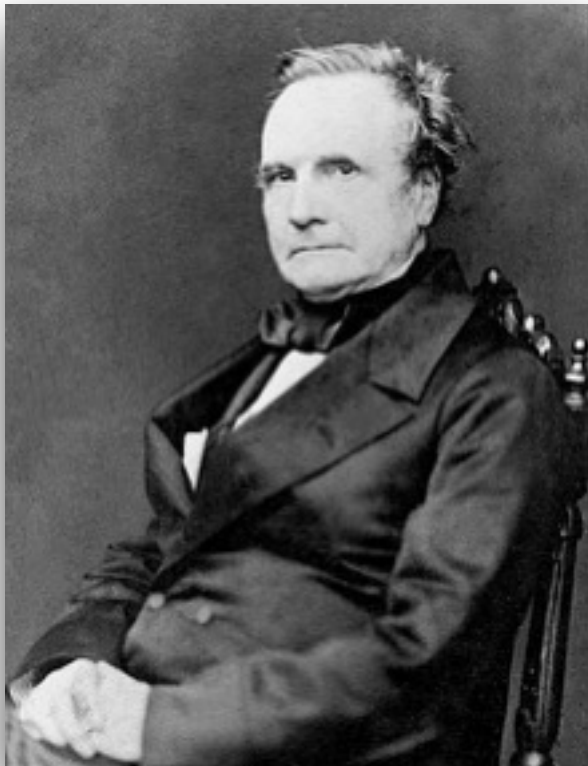
Blaise Pascal
(1623—1662)

Inventor de uma calculadora, a **máquina aritmética** ou **Pascaline**, capaz de somar e subtrair

A máquina aritmética produz efeitos mais próximos ao pensamento do que todas as ações dos animais. Mas ela não faz nada que nos faria atribuir-lhe vontade, como aos animais.

Pensées 340

Charles Babbage



Charles Babbage
1791—1871

- Propôs primeiro computador de uso geral, a **máquina analítica**
- Equivalente em poder computacional a um computador moderno
- Nunca construída

Máquina analítica

Em duas ocasiões foi-me perguntado, “Por favor, senhor Babbage, se colocamos na máquina os números errados, ela produzirá as respostas certas?” (...) Eu não consigo entender que tipo de confusão de idéias poderia provocar uma pergunta dessas.

Charles Babbage

Ada Lovelace



Ada Lovelace
1815—1852

- Considerada primeira programadora
- Escreveu programas para a máquina analítica, que nunca foram executados

Ada Lovelace

É desejável guardar-nos contra a possibilidade de idéias exageradas que possam surgir acerca dos poderes da Máquina Analítica. Ao considerar qualquer nova disciplina, há freqüentemente a tendência de, primeiro, sobrestimar o seu valor; depois, através de uma reação natural, subestimar o verdadeiro valor, quando descobrimos que nossas noções tinham ultrapassado em muito aquilo que era possível.

Ada Lovelace

A Máquina Analítica não tem a pretensão de originar nada. Ela pode fazer tudo que sabemos ordená-la a fazer. Ela pode seguir análise; mas não tem poder de antecipar relações analíticas ou verdades. Seu propósito está em tornar disponível aquilo com o que já estamos familiarizados. (...) ela provavelmente exercerá de outro modo uma influência indireta e recíproca sobre a ciência. Pois, ao combinar as verdades e as fórmulas para análise, de modo que sejam adequadas às combinações mecânicas da máquina, as relações e a natureza de muitas disciplinas são iluminadas e mais profundamente investigadas.

A. Lovelace, Sketch of the Analytical Engine invented by Charles Babbage Esq., By L.F. Manabrea, of Turin, Officer of the Military Engineers, Londres, 1843.

Máquina de Turing (1936)



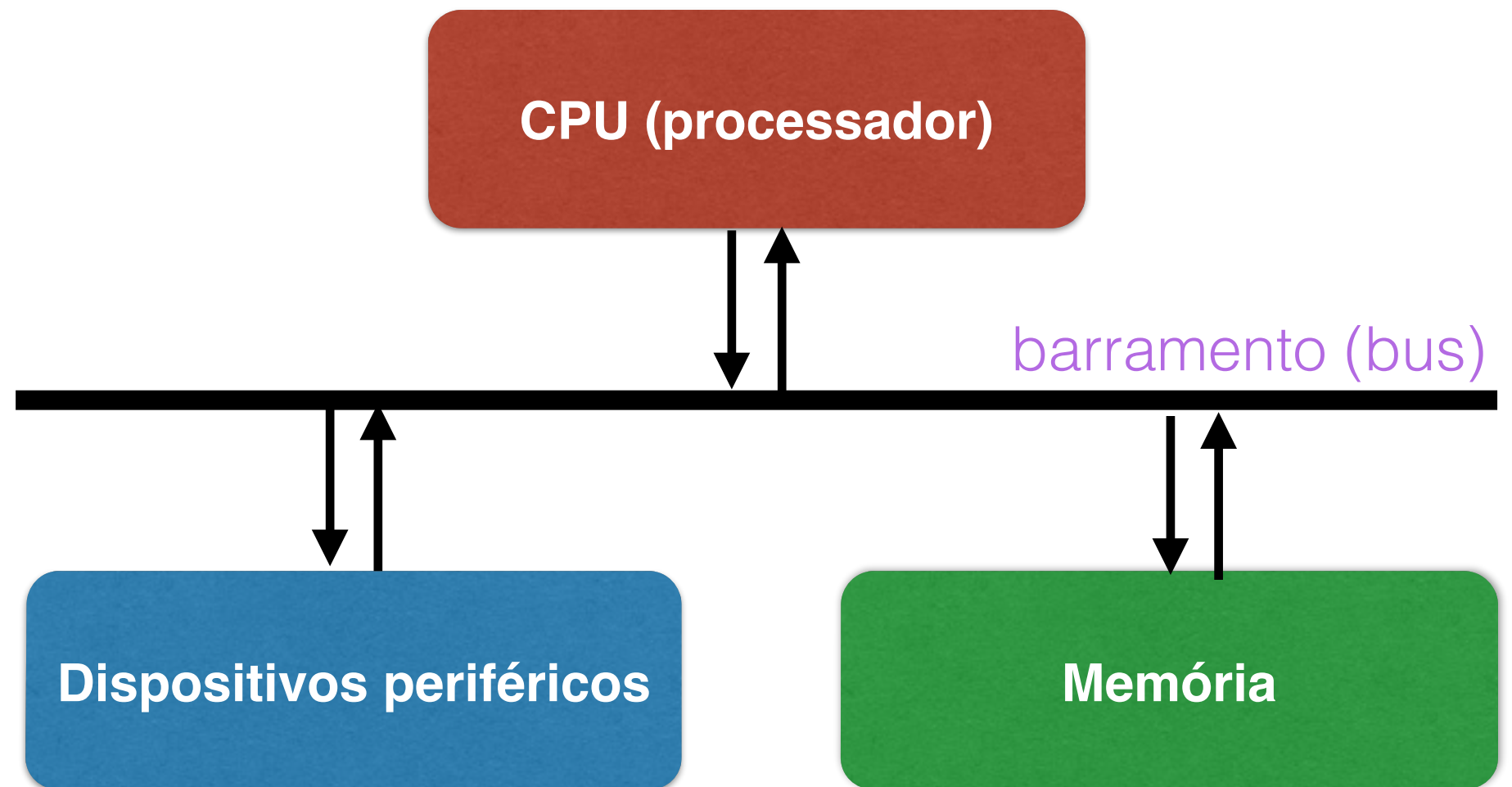
Alan Turing (1912—1954)

- Modelo matemático de computador; definição matemática de algoritmo.
- Máquina universal: máquina capaz de simular qualquer outra máquina, como um computador moderno.

Arquitetura de von Neumann



John von Neumann
(1903—1957)



ENIAC e EDVAC

ENIAC (1946)

- Primeiro computador digital (em notação decimal)
- Descrito como *Giant Brain*
- Programado através de cabos e interruptores; processo longo e complexo

EDVAC (1949)

- Programa guardado na memória
- 5.5 kilobytes de memória (em binário), 1160 somas por segundo, 340 multiplicações por segundo
- Rodou até 1961

Programas na memória

As máquinas eletrônicas daquele tempo (i.e., o Eniac) não possuíam a flexibilidade e generalidade que possuem atualmente para lidar com problemas matemáticos. Cada problema requeria um sistema de cabeamento [wiring] especial e diferente. A grande contribuição de von Neumann foi a idéia de um conjunto de conexões e circuitos fixo e de certa forma universal, um “diagrama de fluxo” e um “código” que permitisse a um conjunto fixo de conexões na máquina ter a capacidade de resolver uma grande variedade de problemas.

S. Ulam, John von Neumann, 1903—1957, Bullentin of the AMS 64, 1958

Elementos da arquitetura

Periféricos

Dispositivos de entrada e saída: monitores, teclados, impressoras, etc.

Memória

Armazena dados. Guarda uma seqüência de *bits*, dígitos 0-1.

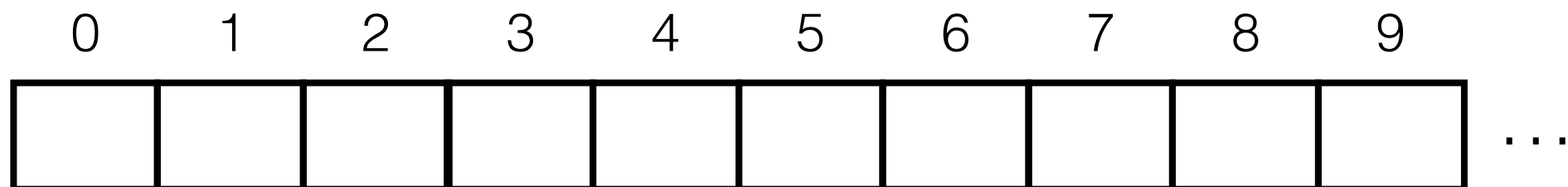
CPU (processador)

Executa instruções que manipulam os dados na memória.

Um computador fictício

Memória

Composta de **células** numeradas a partir do 0. Cada célula guarda 32 bits de informação e pode ser acessada a partir de seu número; seu **endereço**.



Um computador fictício

Processador

Reconhece as seguintes instruções:

| Instrução | Efeito |
|---|---|
| <code>mov <i>num</i>, <i>\$pos</i></code> | Move <i>num</i> para célula <i>pos</i> |
| <code>add <i>\$pos1</i>, <i>\$pos2</i></code> | Soma célula <i>pos1</i> à <i>pos2</i> ; guarda resultado em <i>pos2</i> |
| <code>dec <i>\$pos</i></code> | Subtrai 1 do número em <i>pos</i> |
| <code>jz <i>\$pos</i>, <i>ip</i></code> | Pula para instrução <i>ip</i> se número em <i>pos</i> é ZERO |
| <code>jmp <i>ip</i></code> | Pula para instrução <i>ip</i> |
| <code>nop</code> | <i>No operation</i> : não faz NADA! |

Um pequeno programa

Guardar na célula 500 a soma dos números de 1 a 5.

```
0    mov 5, $40
1    mov 0, $500
2    add $40, $500
3    dec $40
4    jz $40, 6
5    jmp 2
6    nop
```

Linguagem de montagem
(*assembly language*)

Simulação

| Célula 40 | Célula 500 | Instrução |
|-----------|------------|-----------------|
| ? | ? | mov 5, \$40 |
| 5 | ? | mov 0, \$500 |
| 5 | 0 | add \$40, \$500 |
| 5 | 5 | dec \$40 |
| 4 | 5 | jz \$40, 6 |
| 4 | 5 | jmp 2 |
| 4 | 5 | add \$40, \$500 |
| 4 | 9 | dec \$40 |
| 3 | 9 | jz \$40, 6 |
| 3 | 9 | jmp 2 |
| 3 | 9 | add \$40, \$500 |
| 3 | 12 | dec \$40 |
| 2 | 12 | jz \$40, 6 |
| 2 | 12 | jmp 2 |
| 2 | 12 | add \$40, \$500 |
| 2 | 14 | dec \$40 |
| 1 | 14 | jz \$40, 6 |
| 1 | 14 | jmp 2 |
| 1 | 14 | add \$40, \$500 |
| 1 | 15 | dec \$40 |
| 0 | 15 | jz \$40, 6 |
| 0 | 15 | nop |

```
0    mov 5, $40
1    mov 0, $500
2    add $40, $500
3    dec $40
4    jz $40, 6
5    jmp 2
6    nop
```

Desafio

Use as instruções acima para descrever um programa que calcula o fatorial do número contido na célula de memória \$0 e o coloca na célula de memória \$1.