http://www.nickgentry.com/

# Algoritmos e Programação de Computadores

## Disciplina 113476

Prof. Alexandre Zaghetto
http://alexandre.zaghetto.com
zaghetto@unb.br

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

# Prática de Laboratório 01.a
# Algoritmos Sequenciais

# 1. Algoritmos Seqüenciais

**Problema 1a:** Escreva um programa em C que, dado um número de conta corrente com três dígitos fornecido pelo usuário retorna seu dígito verificador utilizando a rotina Módulo 11 [1], [2]. Armazene cada dígito em uma variável diferente.

**Problema 1b:** Escreva a solução para o problema 1a utilizando a linguagem Python.

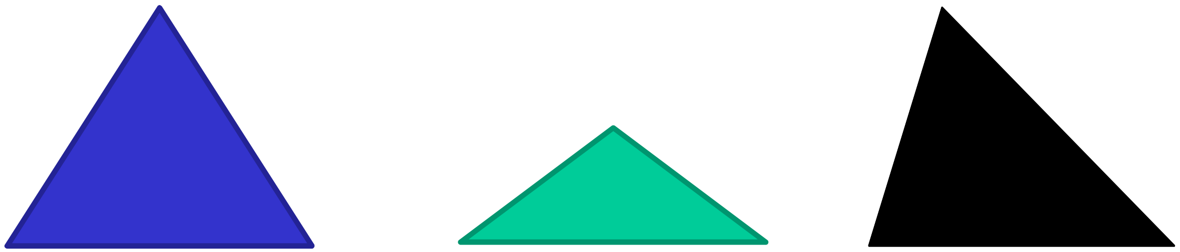
[1] http://pt.wikipedia.org/wiki/D%C3%ADgito_verificador
[2] http://www.cjdinfo.com.br/utilitario-calculo-digito-modulo-11

# Prática de Laboratório 01.b
# Algoritmos com Alternativas

# 1. Algoritmos com Alternativas

**Problema 2a**: Escreva um programa em C que solicita três valores A, B e C ao usuário, e verifica se esses valores satisfazem a condição de existência do triângulo. Caso essa condição seja satisfeita, classifique o triângulo em equilátero, isósceles ou escaleno e escreva a classe do triângulo na tela do computador.

**Problema 2b:** Escreva a solução para o problema 2a utilizando a linguagem Python.



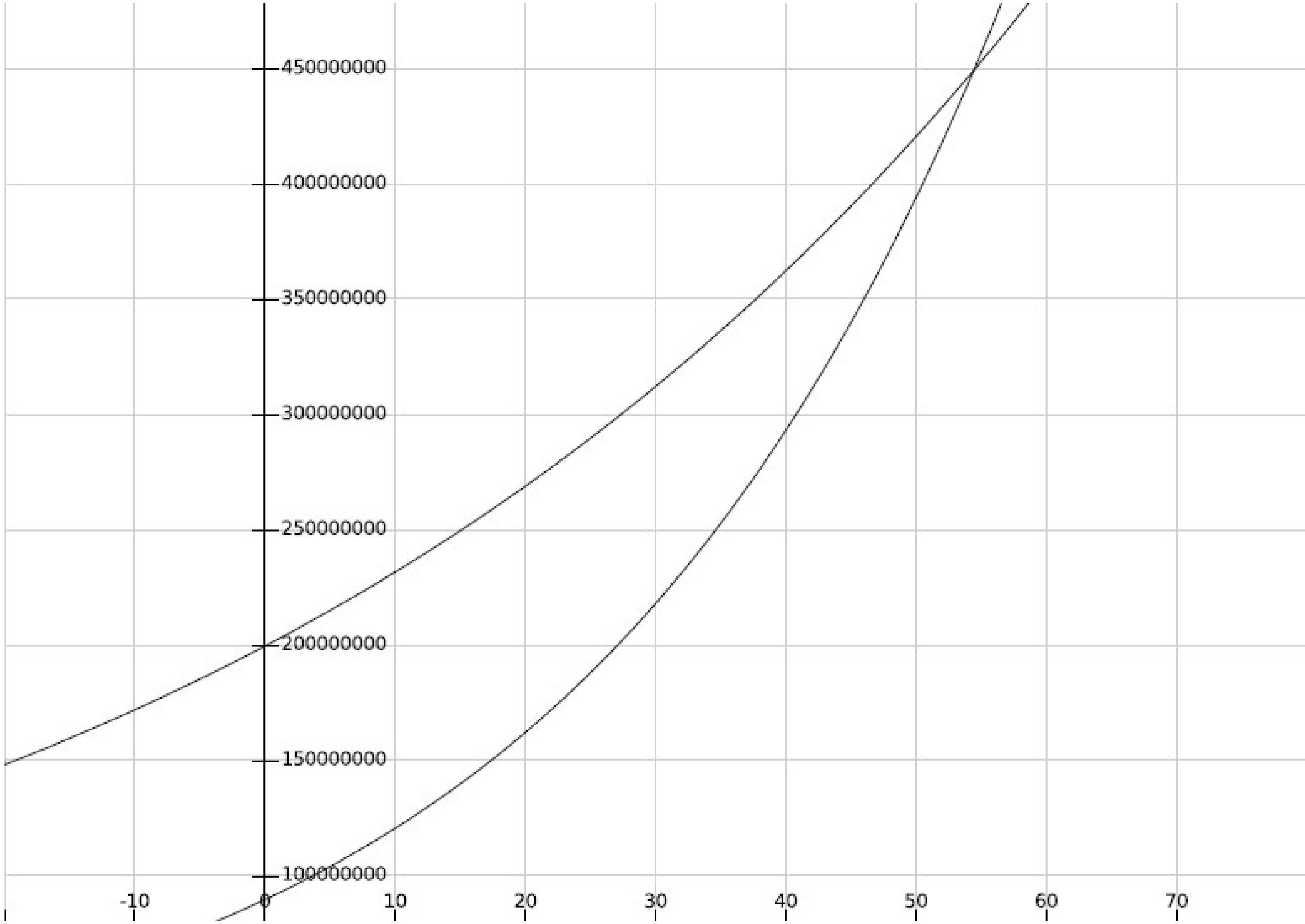
# Prática de Laboratório 01.c
# Algoritmos com Repetição

# 1. Algoritmos com Repetição

**Problema 3a**: Supondo que a população de um país A seja da ordem de 90.000.000 de habitantes com uma taxa anual de crescimento de 3% e que a população de um país B seja, aproximadamente, de 200.000.000 de habitantes com uma taxa anual de crescimento de 1,5%, escreva um programa que calcule numericamente e escreva o número de anos necessários para que a população do país A ultrapasse ou iguale a população do país B, mantidas essas taxas de crescimento.

**Problema 3b:** Escreva a solução para o problema 1a utilizando a linguagem Python.

# 1. Algoritmos com Repetição



# 1. Algoritmos com Repetição

## Problema 4: Raízes de Equações

Em seu curso de álgebra você gastou muita energia para encontrar soluções para equações como

$$7x + 5 = 4x + 3$$

ou

$$x^2 = 3x + 5.$$

# 1. Algoritmos com Repetição

No caso de equações do primeiro grau, escritas na forma $ax + b = 0$, o valor de x é dado por $x = -b/a$. Para equações do segundo grau, escritas na forma $ax^2 + bx + c = 0$, a solução pode ser obtida por meio da fórmula de Bhaskara. A medida que avançamos para uma consideração de equações mais complicadas como

$$3x^3 = 7x + 2$$

ou

$$5\text{sen } x = x + 2$$

ou

$$x^5 = x^4 - 3x^2 + 1,$$

# 1. Algoritmos com Repetição

vemos que fórmulas explícitas para as soluções são tão complicadas que se tornam praticamente inúteis, ou então tais fórmulas nem mesmo existem. Quando se precisa de resposta para esses problemas, somos forçados a lançar mão de aproximações. Existe uma grande abundância de métodos para essas aproximações. Os métodos gráficos são talvez os mais simples dos muitos métodos propostos para encontrar raízes de equações. Suponhamos que você queira obter aproximações para as raízes da equação

$$3x^3 = 7x + 2.$$

# 1. Algoritmos com Repetição

Inicialmente reescrevemos a equação de modo que a expressão à esquerda seja igual a zero,

$$3x^3 - 7x - 2 = 0,$$

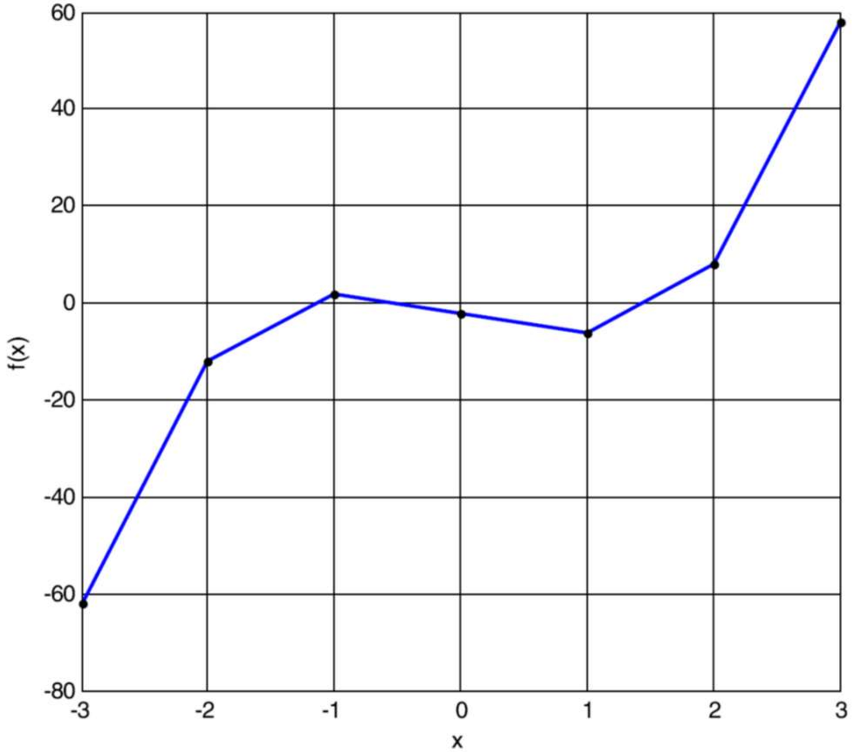
e reformulamos o problema da seguinte maneira: considere a função f(x) dada por,

$$f(x) = 3x^3 - 7x - 2,$$

e ache os valores de x para os quais a função f(x) é igual a zero, ou seja, f(x) = 0. Em seguida traçamos o gráfico de f(x).

# 1. Algoritmos com Repetição

| X | -3 | -2 | -1 | 0 | 1 | 2 | 3 |
|---|---|---|---|---|---|---|---|
| f(x) | -62 | -12 | 2 | -2 | -6 | 8 | 58 |



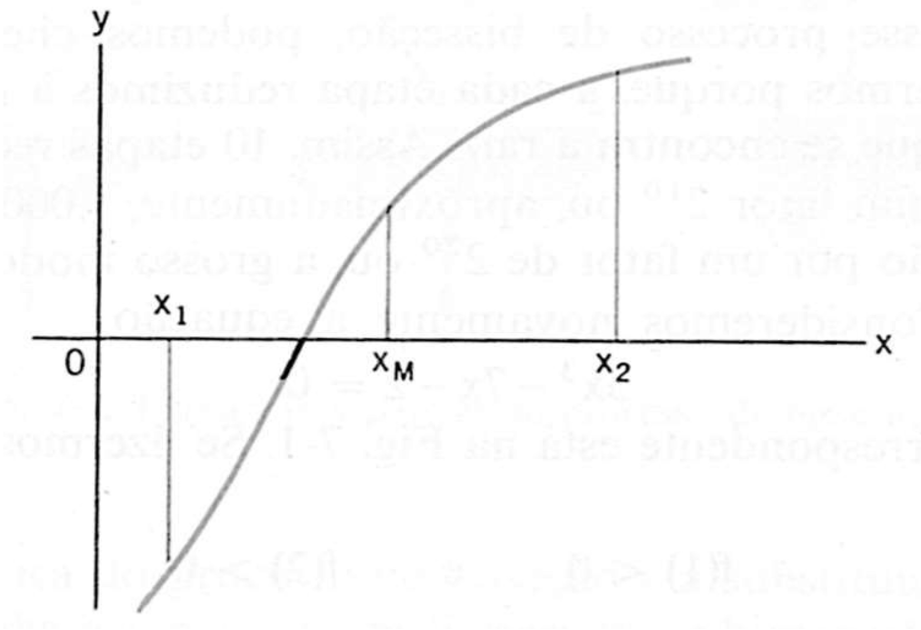
# 1. Algoritmos com Repetição

Podemos verificar a localização das raízes entre -2 e -1, entre -1 e 0 e entre 1 e 2. O método gráfico nos dará uma aproximação para uma raiz da equação. Uma vez que tenhamos uma idéia de onde fica uma raiz, podemos melhorar a sua precisão. Muitos dos métodos empregados em computadores para resolver tais problemas se resumem a métodos de busca. A estratégia geral em métodos de busca é estabelecer que o alvo (neste caso, a raiz de uma equação) deve ser encontrado em algum intervalo de uma variável, e então usar algum teste ou critério para reduzir esse intervalo. O *método da bisseção sucessiva* é uma técnica relativamente simples para reduzir repetidamente o tamanho de um intervalo em que será encontrada uma raiz de uma equação. O método destina-se a ser usado quando se sabe antecipadamente que a função é contínua e tem apenas uma raiz no intervalo.

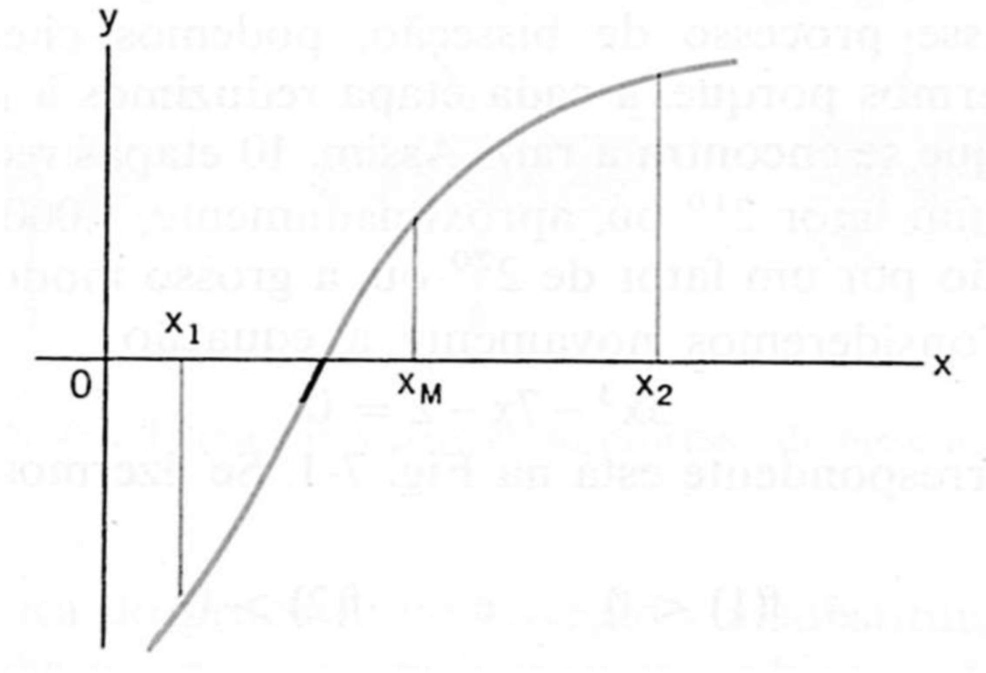
# 1. Algoritmos com Repetição

Suponhamos que estejamos procurando a raiz de uma função f(x) no intervalo [$x_1$ $x_2$]. Suponhamos, ainda, que $f(x_1) < 0$ e $f(x_2) > 0$, isto é, o gráfico de f(x) está abaixo do eixo x quando $x = x_1$, e acima do eixo eixo x quando $x = x_2$. Fazemos a bisseção do intervalo [$x_1$ $x_2$] e marcamos o ponto intermediário $x_M$,

$$x_M = (x_1 + x_2)/2.$$



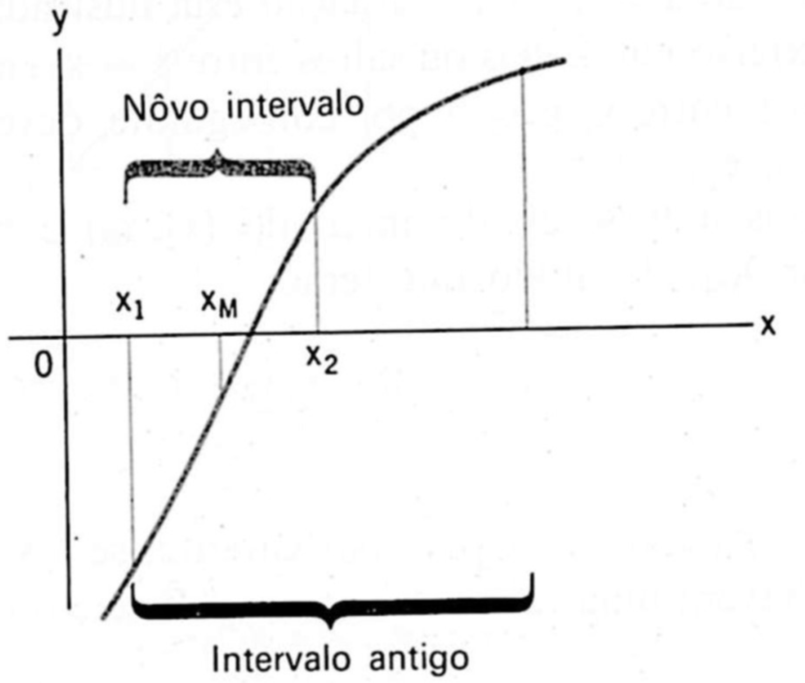
# 1. Algoritmos com Repetição



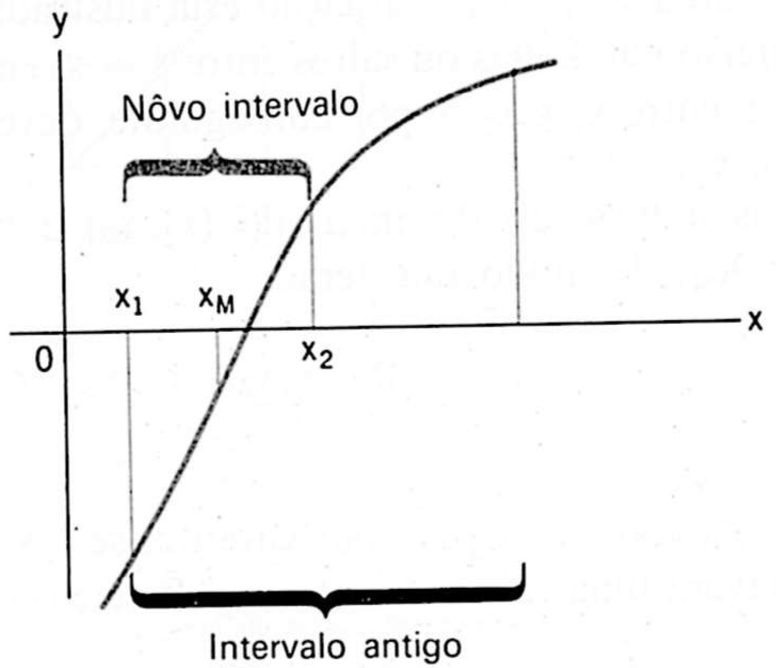

Se $f(x_M) = 0$, então achamos a raiz. No entanto, se $f(x_M) > 0$, então haverá uma raiz **entre $x_1$ e $x_M$**. Esse é o critério para reduzir o tamanho do intervalo de interesse. Atribuímos o valor de $x_M$ a $x_2$ e o processo é repetido, ou seja, calculamos novamente os valores de $x_M$ e $f(x_M)$, agora nesse novo intervalo. Se $f(x_M) < 0$, a raiz está **entre $x_M$ e $x_2$**. Nesse caso, atribuimos o valor de $x_M$ a $x_1$.

# 1. Algoritmos com Repetição

Podemos continuar repetindo este processo, reduzindo o intervalo á metade tantas vezes quanto queiramos, achando, a cada iteração, em que meio intervalo está a raiz.



# 1. Algoritmos com Repetição




A cada etapa, porém, antes de decidirmos prosseguir, podemos verificar a extensão do intervalo, isto é, $|x_2-x_1|$. Se for suficientemente pequeno (digamos menor que um valor $\varepsilon$) aceitamos o valor $x_M$ como sendo a raiz.

# 1. Algoritmos com Repetição

Exemplo:

**Problema**: $3x^4 - 2x^3 + 7x - 4 = 0$ $[0, 1]$ $\in = 0{,}4$
**Solução**: Um número impar de raizes. Para $\in = 0{,}4$ a raiz é 0,625:

$$f(x) = 3x^4 - 2x^3 + 7x - 4 = 0$$

| Etapa | X1 | Sinal de F(X1) | X2 | Sinal de F(X2) | XM | Sinal de F(XM) | \|X1 – X2\| |
|---|---|---|---|---|---|---|---|
| | 0 | – | 1 | + | 0,5 | – | 1 |
| 1 | 0,5 | – | 1 | + | 0,75 | + | 0,5 |
| 2 | 0,5 | – | 0,75 | + | 0.625 | + | 0,25 |

Referência

[1] A. I. Forsythe, T. A. Keenan, E. I. Organick, W. Stenberg, "Ciência de Computadores", Volume 2, Série Ciência de Computação, Ao Livro Técnico S.A., Rio de Janeiro - Guanabara, 1972.

# 1. Algoritmos com Repetição

**Problema 4a**: Implemente o algoritmo da bisseção discutido em sala de aula e determine as raízes das seguintes funções, nos intervalos e com a precisão indicados. Implementar em linguagem C.

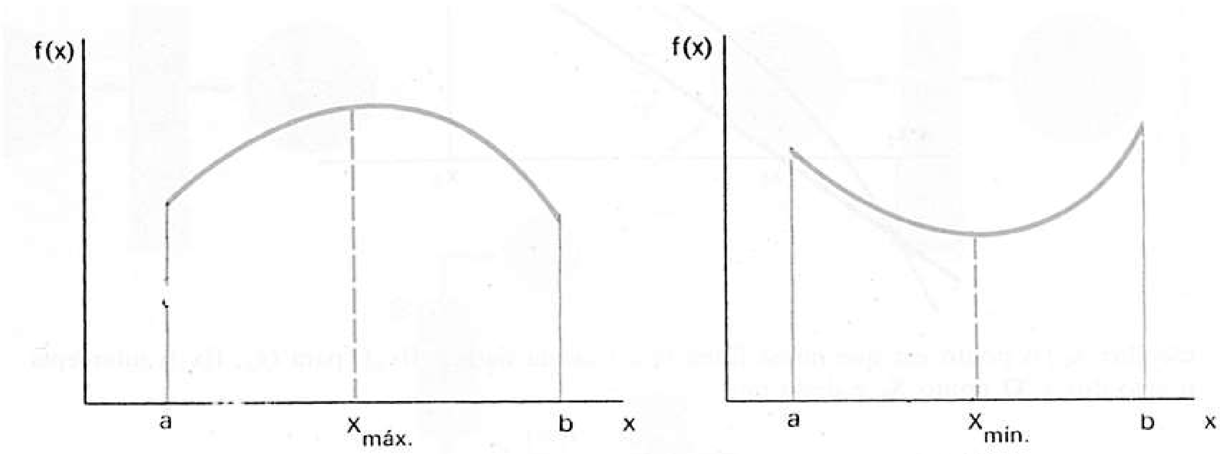
a) $x^3 - x - 1 = 0$, Intervalo [0; 2], $\varepsilon = 0,1$ ➔ Ref.: 1.3247
b) $x + \ln x = 0$, Intervalo [0,1; 1], $\varepsilon = 0,1$ ➔ Ref.: 0.5671
c) $5 - x = 5 * \text{sen } x$, Intervalo [0; 2], $\varepsilon = 0,1$ ➔ Ref.: 0.9456
d) $x^3 - 3x - 2 = 0$, Intervalo [1; 2.5], $\varepsilon = 0,1$ ➔ Ref.: 2
e) $x^3 - 2x^2 - 13x - 10=0$, Intervalo [-1.5; -0.5], $\varepsilon = 0,1$ ➔ Ref.: -1

**Problema 4b**: Escreva a solução para o Problema 4a utilizando a linguagem Python.

# 1. Algoritmos com Repetição

## Problema 5: Máximos e Mínimos

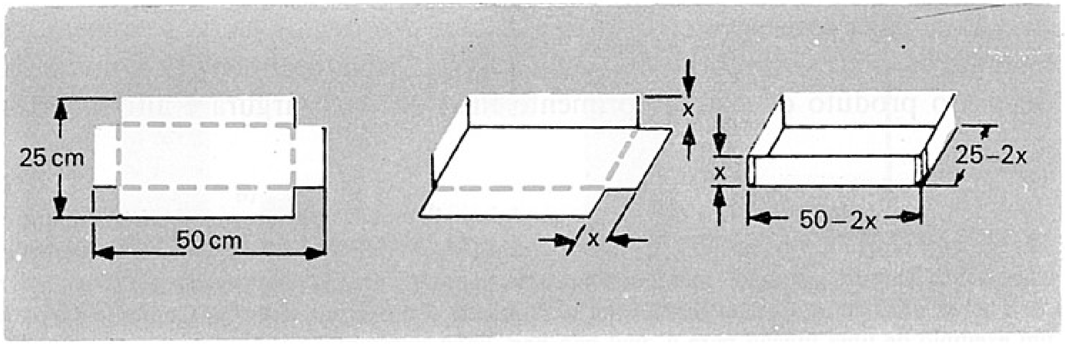
Diz-se que uma função, f(x), tem um *máximo* em um intervalo (a, b) se houver um ponto $x_{max}$ para o qual $f(x_{max})$ é maior do que qualquer outro valor da função no intervalo. A função tem um *mínimo* se houver um ponto $x_{min}$, para o qual $f(x_{min})$ seja menor do que qualquer outro valor da função no intervalo. O termo genérico para máximo e mínimo é *extremos*.




(a) Máximo (b) Minimo

# 1. Algoritmos com Repetição

Suponhamos que você tenha de fazer a maior caixa possível com uma folha de papelão com 50cm de comprimento e 25cm de largura. Um modo de fazer a caixa consiste em cortar quadrados de lado x dos cantos, dobrar as beiradas e juntar os cantos. Naturalmente o volume da caixa é o produto de seu comprimento por sua largura e altura.



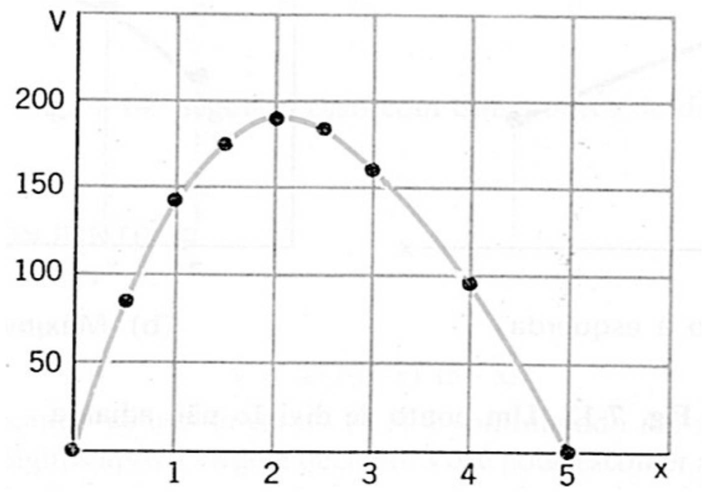

Que tamanho deveriam ter os quadrados cortados (qual deve ser o valor de x) de maneira a se obter o volume máximo?

# 1. Algoritmos com Repetição

Vamos supor que:

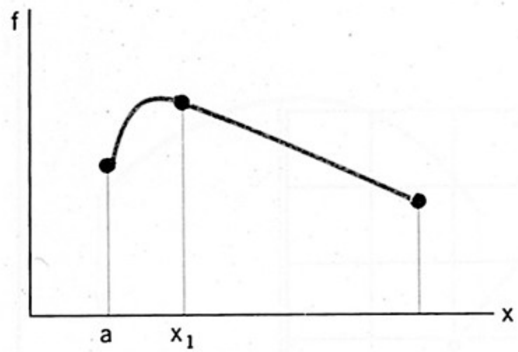
a) a função f(x) tem um único máximo no intervalo (a, b);
b) indo-se da esquerda para a direita, o gráfico de f(x) aumenta progressivamente até o máximo, ou seja, sem intervalos de decréscimos temporários;
c) indo mais ainda para a direita, f(x) decresce progressivamente a partir do máximo, sem intervalos de acréscimos temporários.

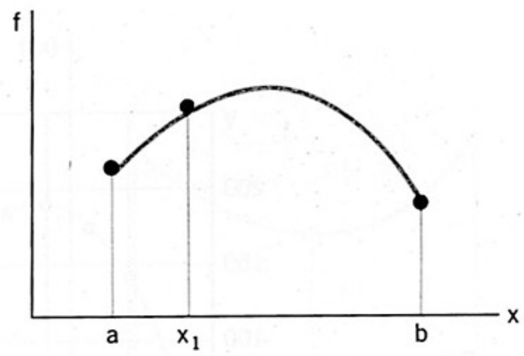
Em outras palavras, f(x) teria o aspecto abaixo.



# 1. Algoritmos com Repetição

Para buscar o máximo de uma função f(x) é razoável avaliar a função em um ponto $x_1$, dentro de (a, b). O que podemos aprender de $f(x_1)$? Podemos sempre encontrar funções que têm seu máximo em qualquer lado de $x_1$, e continuaremos como antes, sem saber onde está o máximo.

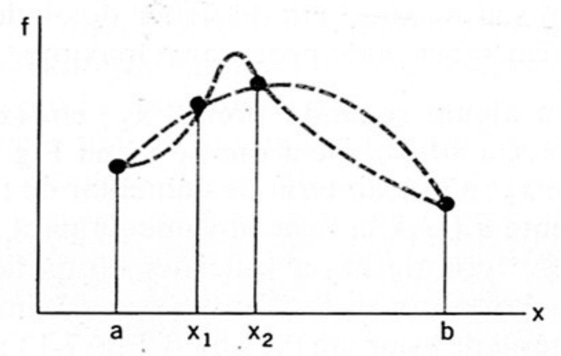



É igualmente razoável avaliar a função em algum segundo ponto $x_2$ em $(x_1, b)$.

# 1. Algoritmos com Repetição

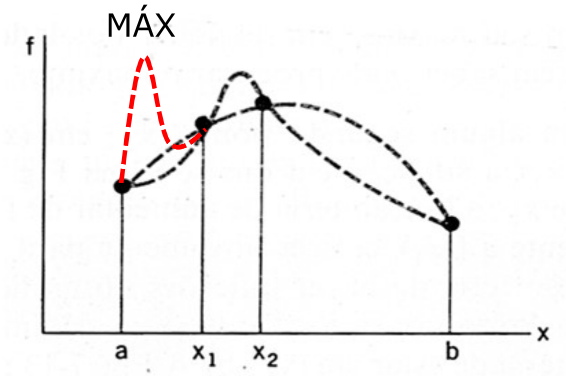
Se $f(x_2) > f(x_1)$, temos as seguintes possíveis situações:




Supondo que o máximo estivesse entre a e $x_1$, a função teria de aumentar a partir de f(a) até seu máximo, decrescer temporariamente até $f(x_1)$, crescer novamente para $f(x_2)$ e finalmente decrescer para f(b). Então teria de haver um intervalo de acréscimo temporário entre o máximo e f(b), o que fere a suposição (c). Logo, uma vez que $f(x_2)>f(x_1)$, máximo não pode estar entre a e $x_1$, só podendo estar em $(x_1, b)$.

# 1. Algoritmos com Repetição

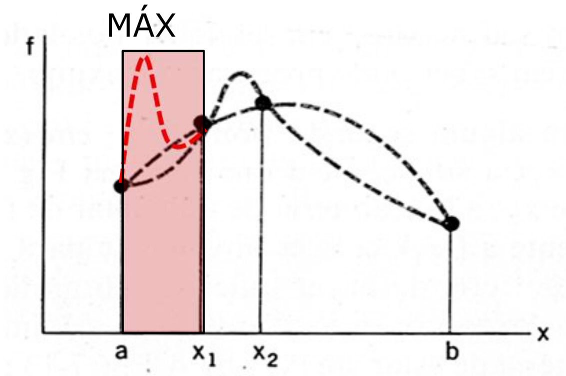
Se $f(x_2) > f(x_1)$, temos as seguintes possíveis situações:




Supondo que o máximo estivesse entre a e $x_1$, a função teria de aumentar a partir de f(a) até seu máximo, decrescer temporariamente até $f(x_1)$, crescer novamente para $f(x_2)$ e finalmente decrescer para f(b). Então teria de haver um intervalo de acréscimo temporário entre o máximo e f(b), o que fere a suposição (c). Logo, uma vez que $f(x_2)>f(x_1)$, máximo não pode estar entre a e $x_1$, só podendo estar em $(x_1, b)$.

# 1. Algoritmos com Repetição

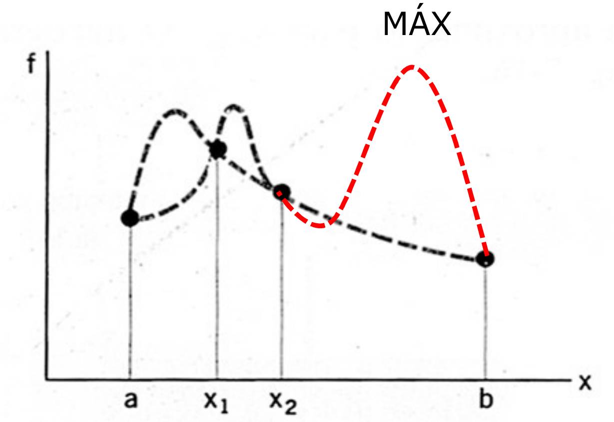
Se $f(x_2) > f(x_1)$, temos as seguintes possíveis situações:




Supondo que o máximo estivesse entre a e $x_1$, a função teria de aumentar a partir de f(a) até seu máximo, decrescer temporariamente até $f(x_1)$, crescer novamente para $f(x_2)$ e finalmente decrescer para f(b). Então teria de haver um intervalo de acréscimo temporário entre o máximo e f(b), o que fere a suposição (c). Logo, uma vez que $f(x_2)>f(x_1)$, máximo não pode estar entre a e $x_1$, só podendo estar em $(x_1, b)$.

# 1. Algoritmos com Repetição

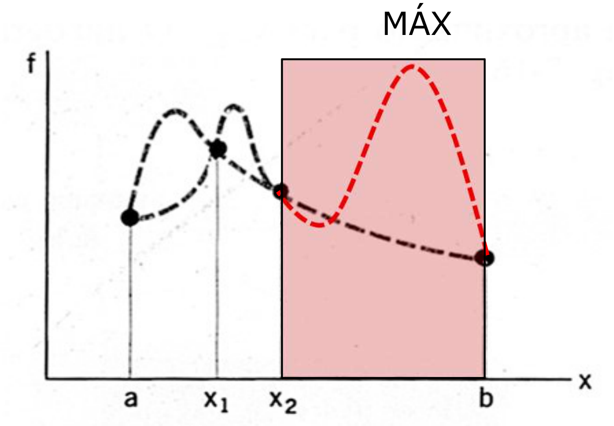
Se $f(x_2) < f(x_1)$, temos as seguintes possíveis situações:




Agora, supondo que o máximo estivesse no intervalo $(x_2, b)$, a função teria que aumentar de a até $f(x_1)$, decrescer temporariamente até $f(x_2)$, crescer novamente até seu máximo e finalmente decrescer até f(b). Então teria de haver um intervalo de decrescimento temporário entre f(a) e o máximo, o que fere a suposição (b). Logo, uma vez que $f(x_2)<f(x_1)$, o máximo não pode estar entre $x_2$ e b, só podendo estar em $(a, x_2)$.

# 1. Algoritmos com Repetição

Se $f(x_2) < f(x_1)$, temos as seguintes possíveis situações:




Agora, supondo que o máximo estivesse no intervalo $(x_2, b)$, a função teria que aumentar de a até $f(x_1)$, decrescer temporariamente até $f(x_2)$, crescer novamente até seu máximo e finalmente decrescer até f(b). Então teria de haver um intervalo de decrescimento temporário entre f(a) e o máximo, o que fere a suposição (b). Logo, uma vez que $f(x_2)<f(x_1)$, o máximo não pode estar entre $x_2$ e b, só podendo estar em $(a, x_2)$.

# 1. Algoritmos com Repetição

Uma vez tendo sido limitado o intervalo de busca, podemos continuar repetindo este processo, reduzindo o intervalo tantas vezes quanto queiramos, achando a cada iteração em que parte do intervalo o máximo está localizado. A cada etapa, porém, antes de decidirmos prosseguir, podemos verificar a extensão do intervalo, isto é, $|b-a|$. Se for suficientemente pequeno (digamos menor que um valor $\varepsilon$) aceitamos o valor $(b+a)/2$ como sendo o máximo.

Uma forma conveniente de se escolher os valores de $x_1$ e $x_2$ é o Método da Trissecção, que divide o intervalo (a, b) em suas terças partes, ou seja, $x_1 = a+(b-a)/3$ e $x_2 = b-(b-a)/3$.

Uma análise análoga pode ser utilizada na procura por um mínimo.

# 1. Algoritmos com Repetição

**Problema 5a**: Considerando o roteiro acima, escreva um programa em C que determina qual tamanho devem ter os quadrados cortados (qual deve ser o valor de x) de maneira a se obter o volume máximo para a caixa.

**Problema 5b**: Escreva a solução para o Problema 5a utilizando a linguagem Python.