**Teorema Fundamental da Álgebra (informal)**. Um polinômio de grau $n$ tem exatamente $n$ raízes.

# Algoritmos Aleatorizados - Corretude

Prof. André Vignatti

Dados $F(x)$ e $G(x)$ dois polinômios de grau $d$ dados como:

$$F(x) = (x+1)(x-2)(x+3)(x-4)(x+5)(x-6)$$
$$G(x) = x^6 - 7x^3 + 25$$

Como saber se $F(x) \equiv G(x)$?

- Solução natural em $O(d^2)$.

Considere o seguinte algoritmo aleatorizado:

ALGORITMO VP

1. escolha $r \in \{1, \ldots, 100d\}$ aleatoriamente de maneira uniforme.
2. verifique se $F(r)$ é igual a $G(r)$. `(tempo` $O(d)$`)`
3. se $F(r) = G(r)$, devolve SIM; `(sem sempre acerta)`
4. caso contrário, devolve NÃO. `(100% certo)`

VP executa em tempo $O(d)$. O algoritmo erra?

**Exemplo.** Sejam $F(x) = x^2$ e $G(x) = x^2 - 2x + 4$. Claramente $F(x) \neq G(x)$. Mas $F(2) = G(2)$.

Quando o algoritmo erra?

- Erra quando $r$ é raiz de $H(x) = 0$, onde $H(x) = F(x) - G(x)$. `(o alg.acha que` $F(x) = G(x)$`, mas na verdade` $F(x) \neq G(x)$`)`

$H(x)$ tem grau $\leq d \Longrightarrow H(x)$ tem $\leq d$ raízes. Assim,

$$\Pr(\textit{VP errar}) \leq \frac{d}{100d} = \frac{1}{100}$$

**Probabilidade é traiçoeira!** Exemplo: uma mulher está grávida de gêmeos. Após o exame, soube que um é menina. Qual a probabilidade de ter 2 meninas? **Portanto, antes de tudo**: precisamos formalizar nossa conversa.

**Definição.** *Um* ***espaço de probabilidade discreto*** *tem 3 componentes:*

- conjunto $\Omega$, chamado de *espaço amostral*
- conjunto $\mathcal{F}$ de todos subconjuntos de $\Omega$, cada $E \in \mathcal{F}$ é chamado de *evento.*
- função de probabilidade $\Pr : \mathcal{F} \to \mathbb{R}^+$

**Exemplo.** Se $\Omega = \{\bigstar, \blacksquare, \spadesuit\}$ então

$$\mathcal{F} = \Big\{\emptyset, \{\bigstar\}, \{\blacksquare\}, \{\spadesuit\}, \{\bigstar, \blacksquare\}, \{\bigstar, \spadesuit\}, \{\blacksquare, \spadesuit\}, \{\bigstar, \blacksquare, \spadesuit\}\Big\}$$

E $\{\bigstar, \blacksquare\}$ é exemplo de um evento.

**Definição** (**Axiomas de Kolmogorov**)**.** *Uma* função de probabilidade *é uma função* $\Pr : \mathcal{F} \to \mathbb{R}^+$ *tal que*

- $0 \leq \Pr(E) \leq 1, \ \forall E \in \mathcal{F}$
- $\Pr(\Omega) = 1$
- Seja $E_1, E_2, \ldots,$ eventos **disjuntos**. Então

$$\Pr(E_1 \cup E_2 \ldots) = \Pr(E_1) + \Pr(E_2) + \ldots$$

**Exemplo.** Na verificação de polinômios

- $\Omega = \{1, \ldots, 100d\}$
- Cada escolha de $r = i$ é o evento simples $E_i = \{i\}$
- $r$ é escolhido uniformemente $\implies \Pr(E_i) = \Pr(E_j), \ \forall i, j.$
- $\Pr(\Omega) = 1 \Rightarrow \Pr(E_i) = \frac{1}{100d}$ (pois $\bigcup_{i \geq 1} E_i = \Omega$).

**Exemplo.** Considere o lance de um dado de 6 lados.

- $\Omega = \{\boxdot, \boxdot, \boxdot, \boxdot, \boxdot, \boxdot\}$.

Exemplo de eventos que podemos considerar

- $E'$ = evento do dado mostrar número par = $\{\boxdot, \boxdot, \boxdot\}$.
- $E''$ = evento do dado mostrar número menor que 3 = $\{\boxdot, \boxdot\}$.
- $E'''$ = evento do dado mostrar número primo= $\{\boxdot, \boxdot, \boxdot\}$.

**Lema.** *Para eventos $E_1$ e $E_2$ temos*

$$\Pr(E_1 \cup E_2) = \Pr(E_1) + \Pr(E_2) - \Pr(E_1 \cap E_2)$$

*Demonstração.* (Só dar ideia com uma figura) □

**Corolário.** *Para eventos $E_1$ e $E_2$ temos*

$$\Pr(E_1 \cup E_2) \leq \Pr(E_1) + \Pr(E_2)$$

**Teorema (Limitante da União).** *Dados eventos $E_1, E_2, \ldots$ temos*

$$\Pr\left(\bigcup_{i\geq 1} E_i\right) \leq \sum_{i\geq 1} \Pr(E_i)$$

**Suposição computacional**: obter um número (pseudo) aleatório leva tempo $\Theta(1)$.

Método popular: **Gerador de Congruência Linear**

- é uma recorrência: $f(n) = (af(n-1) + c) \mod m$,
- $f(0)$ é chamado de *semente* (seed)
- em linguagem `C`: $a = 1103515245, c = 12345, m = 2^{31}$.
- em linguagem `Java`: $a = 25214903917, c = 11, m = 2^{48}$.

**Definição.** *Dois eventos $E$ e $F$ são ditos serem **independentes** sse*

$$\Pr(E \cap F) = \Pr(E) \cdot \Pr(F)$$

Como diminuir a probabilidade de erro para $\frac{1}{\text{1bilhão}}$ ?

- 1ª tentativa: aumentar o espaço amostral
  - faixa de valores limitada pela precisão da máquina
  - sorteio do $r$ pode não levar tempo constante!
- 2ª tentativa: executar várias vezes o algoritmo

ALGORITMO $\text{VP}_k$

1. execute o algoritmo $VP$ $k$ vezes (com reposição).
2. devolve NÃO se em uma das $k$ execuções o $VP$ devolve não;
3. caso contrário, devolve SIM.

- Seja $E_i$ o evento do algoritmo escolher raiz de $F(x) - G(x) = 0$ na i-ésima execução de $VP$.

- Os eventos $E_i$ são mutuamente independentes.
- A probabilidade do algoritmo falhar é:

$$\Pr(E_1 \cap E_2 \cap \ldots \cap E_k) = \prod_{i=1}^{k} \Pr(E_i) \leq \prod_{i=1}^{k} \frac{d}{100d} \leq \left(\frac{1}{100}\right)^k$$

**Filosofando...**

- Parece errado um algoritmo que pode dar a resposta errada!

**Estamos acostumados com**:

- algoritmos 100% corretos.
- otimizar o tempo de execução.
- sacrificar tempo de execução gastanto menos memória.

Abra sua cabeça! **Porque não pensar em**:

- algoritmo $<$ 100% corretos?
- otimizar a corretude?
- sacrificar tempo de execução aumentando a corretude?