Algoritmos Randomizados: Introdução

Celina Figueiredo
Guilherme Fonseca
Manoel Lemos
Vinícius Sá

impa

26º Colóquio Brasileiro de Matemática
IMPA – Rio de Janeiro – Brasil
2007

Resumo

- Definições
- Monte Carlo
- Variáveis Aleatórias
- Las Vegas
- Paradigmas combinatórios
- Método probabilístico

Definições

- Algoritmo
- Experimento aleatório (ou randômico)
- Gerador de números aleatórios
- Algoritmos randomizados

Algoritmos Randomizados

- Aplicações
  - criptografia
  - programação distribuída
  - teoria dos grafos
  - geometria computacional
  - etc.

- Vantagens
  - mais rápidos
  - mais simples
  - ambos

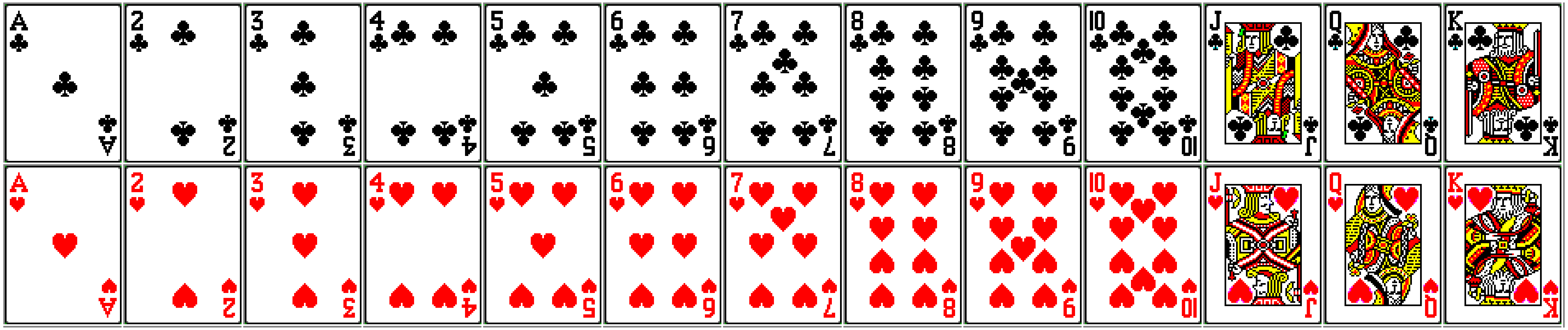
- Preço
  - análise trabalhosa
  - <u>incerteza</u>
    - qualidade da resposta
    - tempo de execução

Monte Carlo

- Fornecem a resposta correta com probabilidade (alta) conhecida
- Tempo de execução determinístico

Las Vegas

- A resposta dada está sempre correta
- Tempo de execução é uma variável aleatória



Algoritmos de Monte Carlo

(p / problemas de decisão)

- Erro bilateral

OU

- Erro unilateral
  - baseados-no-SIM
  - baseados-no-NÃO

Algoritmos de Monte Carlo

C_N: a resposta correta é NÃO

C_S: a resposta correta é SIM

A_N: o algoritmo responde NÃO

A_S: o algoritmo responde SIM

baseado-no-NÃO: $\mathbf{Pr}\,\{ C_N \mid A_N \} = 1$

$\mathbf{Pr}\,\{ A_S \mid C_S \} = 1$

Algoritmos de Monte Carlo

(baseados-no-NÃO)

$\mathbf{Pr}$ {"erro"}

$= \mathbf{Pr} \{ C_N , A_S \;\; \text{U} \;\; C_S , A_N \}$

$= \mathbf{Pr} \{ C_N , A_S \} + \mathbf{Pr} \{ C_S , A_N \}$

$= \mathbf{Pr} \{ C_N \} \, . \, \mathbf{Pr} \{ A_S \mid C_N \} + \mathbf{Pr} \{ C_S \} \, . \, \mathbf{Pr} \{ A_N \mid C_S \}$

Algoritmos de Monte Carlo
(baseados-no-NÃO)

$\mathbf{Pr}$ {“erro”}

$= \mathbf{Pr} \{ C_N , A_S \ \cup \ C_S , A_N \}$

$= \mathbf{Pr} \{ C_N , A_S \} + \mathbf{Pr} \{ C_S , A_N \}$

$= \mathbf{Pr} \{ C_N \} . \mathbf{Pr} \{ A_S | C_N \} + \mathbf{Pr} \{ C_S \} . \mathbf{Pr} \{ A_N | C_S \}$

$= \mathbf{Pr} \{ C_N \} . \mathbf{Pr} \{ A_S | C_N \} + \mathbf{Pr} \{ C_S \} . 0$

Algoritmos de Monte Carlo

(baseados-no-NÃO)

$\mathbf{Pr}$ {“erro”}

$= \mathbf{Pr}\{ C_N , A_S \;\; U \;\; C_S , A_N \}$

$= \mathbf{Pr}\{ C_N , A_S \} + \mathbf{Pr}\{ C_S , A_N \}$

$= \mathbf{Pr}\{ C_N \} . \mathbf{Pr}\{ A_S \mid C_N \} + \mathbf{Pr}\{ C_S \} . \mathbf{Pr}\{ A_N \mid C_S \}$

$= \mathbf{Pr}\{ C_N \} . \mathbf{Pr}\{ A_S \mid C_N \} + \mathbf{Pr}\{ C_S \} . 0$

$= \mathbf{Pr}\{ C_N \} . \quad \boldsymbol{\varepsilon} \quad + 0$

Algoritmos de Monte Carlo
(baseados-no-NÃO)

$\mathbf{Pr}$ {“erro”}

$= \mathbf{Pr}\{ C_N , A_S \;\; U \;\; C_S , A_N \}$

$= \mathbf{Pr}\{ C_N , A_S \} + \mathbf{Pr}\{ C_S , A_N \}$

$= \mathbf{Pr}\{ C_N \} . \mathbf{Pr}\{ A_S \mid C_N \} + \mathbf{Pr}\{ C_S \} . \mathbf{Pr}\{ A_N \mid C_S \}$

$= \mathbf{Pr}\{ C_N \} . \mathbf{Pr}\{ A_S \mid C_N \} + \mathbf{Pr}\{ C_S \} . 0$

$= \mathbf{Pr}\{ C_N \} . \;\; \boldsymbol{\varepsilon} \;\; + 0$

$\leq \boldsymbol{\varepsilon}$

Algoritmos de Monte Carlo

(baseados-no-NÃO)

$\mathbf{Pr}$ {"erro"}

$= \mathbf{Pr} \{ C_N , A_S \;\; U \;\; C_S , A_N \}$

$= \mathbf{Pr} \{ C_N , A_S \} + \mathbf{Pr} \{ C_S , A_N \}$

$= \mathbf{Pr} \{ C_N \} . \mathbf{Pr} \{ A_S \mid C_N \} + \mathbf{Pr} \{ C_S \} . \mathbf{Pr} \{ A_N \mid C_S \}$

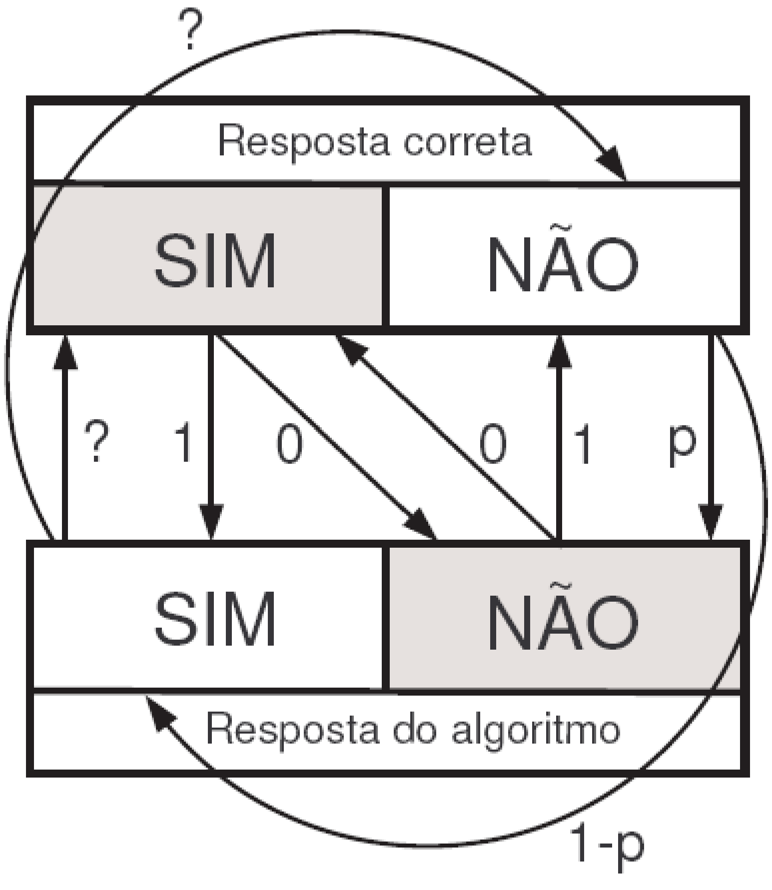
$= \mathbf{Pr} \{ C_N \} . \mathbf{Pr} \{ A_S \mid C_N \} + \mathbf{Pr} \{ C_S \} . 0$

$= \mathbf{Pr} \{ C_N \} . \quad \boldsymbol{\varepsilon} \quad + 0$

$\leq \boldsymbol{\varepsilon} \longrightarrow$ $\boxed{\mathbf{Pr}\{\text{“acerto”}\} \geq \boldsymbol{p} = \mathbf{1} - \boldsymbol{\varepsilon}}$

Algoritmos de Monte Carlo
(baseados-no-NÃO)

- Quando respondem NÃO, estão sempre corretos (exibem certificado)



Algoritmos de Monte Carlo
(baseados-no-NÃO)

Exemplo: IDENTIDADE DE POLINÔMIOS

$$F(x) = (x - a_1)\,(x - a_2)\,\ldots\,(x - a_d)$$

$$G(x) = b_d x^d + b_{d-1} x^{d-1} + \ldots + b_1 x + b_0$$

<u>Determinístico</u>

1) Transforme $F(x)$
2) Compare os coeficientes de $F(x)$ e $G(x)$
3) Se houver diferença, retorne NÃO
4) Senão, retorne SIM

<u>Monte Carlo</u>

1) Sorteie um inteiro w, aleatoriamente, de 1 a $100d$
2) Avalie $F(w)$ e $G(w)$
3) Se $F(w) \neq G(w)$, retorne NÃO
4) Senão, retorne SIM

Algoritmos de Monte Carlo
(baseados-no-NÃO)

Exemplo: IDENTIDADE DE POLINÔMIOS

$$F(x) = (x - a_1)(x - a_2) \dots (x - a_d)$$

$$G(x) = b_d x^d + b_{d-1} x^{d-1} + \dots + b_1 x + b_0$$

Determinístico → $\mathbf{O(d^2)}$

1) Transforme $F(x)$
2) Compare os coeficientes de $F(x)$ e $G(x)$
3) Se houver diferença, retorne NÃO
4) Senão, retorne SIM

Monte Carlo

1) Sorteie um inteiro w, aleatoriamente, de 1 a $100d$
2) Avalie $F(w)$ e $G(w)$
3) Se $F(w) \neq G(w)$, retorne NÃO
4) Senão, retorne SIM

Algoritmos de Monte Carlo
(baseados-no-NÃO)

Exemplo: IDENTIDADE DE POLINÔMIOS

$$F(x) = (x - a_1)(x - a_2) \ldots (x - a_d)$$

$$G(x) = b_d x^d + b_{d-1} x^{d-1} + \ldots + b_1 x + b_0$$

<u>Determinístico</u> → $\mathbf{O(d^2)}$

1) Transforme $F(x)$
2) Compare os coeficientes de $F(x)$ e $G(x)$
3) Se houver diferença, retorne NÃO
4) Senão, retorne SIM

<u>Monte Carlo</u> → $\mathbf{O(d)}$

1) Sorteie um inteiro w, aleatoriamente, de 1 a $100d$
2) Avalie $F(w)$ e $G(w)$
3) Se $F(w) \neq G(w)$, retorne NÃO
4) Senão, retorne SIM

Algoritmos de Monte Carlo
(baseados-no-NÃO)

Exemplo: IDENTIDADE DE POLINÔMIOS

$$F(x) = (x - a_1)\,(x - a_2)\,\ldots\,(x - a_d)$$

$$G(x) = b_d x^d + b_{d-1} x^{d-1} + \ldots + b_1 x + b_0$$

Pr $\{ A_N \mid C_S \} = 0$

Pr $\{ A_S \mid C_N \} = \boldsymbol{\varepsilon} = ?$

$\boldsymbol{\varepsilon} \leq d \,/\, 100d = 1/100$

Pr {"acerto"} $\geq$ 99%

<u>Monte Carlo</u> → **O(*d*)**

1) Sorteie um inteiro *w*, aleatoriamente, de 1 a 100*d*
2) Avalie $F(w)$ e $G(w)$
3) Se $F(w) \neq G(w)$, retorne NÃO
4) Senão, retorne SIM

Algoritmos de Monte Carlo
(baseados-no-NÃO)

Refinando a probabilidade de acerto...

- Em uma execução do algoritmo,

  $\mathbf{Pr}$ {“erro”} $\leq$ $\mathbf{Pr}$ $\{ A_S \mid C_N \}$ $= \boldsymbol{\varepsilon}$

- Em t execuções independentes,

  $\mathbf{Pr}$ {“erro”} $=$ $\mathbf{Pr}$ {“erro$_1$”,“erro$_2$”, ... ,“erro$_t$”} $\leq \boldsymbol{\varepsilon}^t$

Variáveis Aleatórias

- Função que mapeia um experimento aleatório em um valor numérico qualquer

$$X : \Omega \rightarrow \mathbb{R}$$

A = soma dos valores obtidos no lançamento de dois dados

B = número de sorteios até que se complete determinada coluna de uma cartela de bingo

$$C = \begin{cases} 1 \text{ , se cara} \\ 0 \text{ , se coroa} \end{cases}$$

Variáveis Aleatórias

- Esperança (ou valor esperado)

  média dos resultados possíveis ponderada pelas probabilidades de ocorrência

A = soma dos valores obtidos no lançamento de dois dados

$$
\begin{aligned}
\mathbf{E}\,[A] \;=\;& \quad 2 \,.\, \mathbf{Pr}\,\{\text{“}A = 2\text{”}\} \;+ \\
&+\; 3 \,.\, \mathbf{Pr}\,\{\text{“}A = 3\text{”}\} \;+ \\
&+\; \ldots \;+ \\
&+\; 12 \,.\, \mathbf{Pr}\,\{\text{“}A = 12\text{”}\} \;= \\
=\;& 7
\end{aligned}
$$

Variáveis Aleatórias

- Esperança: $\mathbf{E}\,[X] = \Sigma\,(\,j\,.\,\mathbf{Pr}\,\{\text{“}X = j\text{”}\}\,)$
- Variância: $\mathbf{Var}\,[X] = \mathbf{E}\,[X^2] - (\mathbf{E}\,[X])^2$
- Desvio padrão
- Momentos da V. A.

Variáveis Aleatórias famosas

$$\mathbf{Pr}\{\text{“sucesso”}\} = p$$

- <u>Bernoulli</u>

$$X = \begin{cases} 1, \text{ se “sucesso”} \\ 0, \text{ se “fracasso”} \end{cases}$$

$\mathbf{E}[X] = p$

$\mathbf{Var}[X] = p(1-p)$

- <u>Binomial</u>

X = “número de sucessos em n experimentos independentes”

$\mathbf{E}[X] = n\,p$

$\mathbf{Var}[X] = n\,p\,(1-p)$

- <u>Geométrica</u>

X = “número de experimentos até o primeiro sucesso”

$\mathbf{E}[X] = 1 / p$

$\mathbf{Var}[X] = (1-p) / \mathrm{p}^2$

Desigualdades famosas

- Desigualdade de Markov

$$\mathbf{Pr}\{\, X \geq a \,\} \leq \frac{\mathbf{E}[X]}{a} \qquad (a > 0)$$

- Desigualdade de Chebyshev

$$\mathbf{Pr}\{\, |X - \mathbf{E}[X]| \geq a \,\} \leq \frac{\mathbf{Var}[X]}{a^2} \qquad (a > 0)$$

Algoritmos de Las Vegas

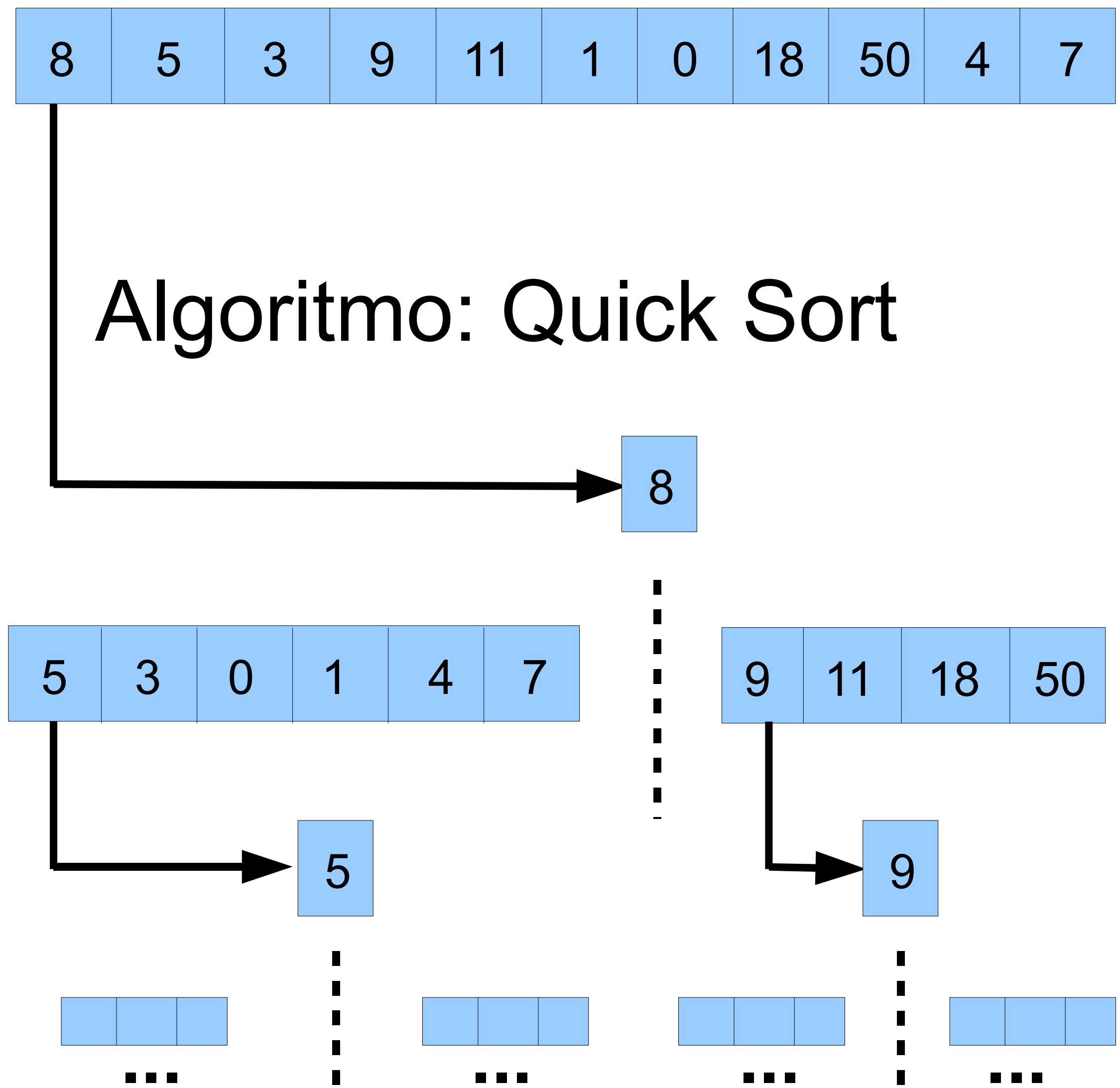
- Resposta <u>sempre</u> correta

- Tempo computacional é uma V. A.

Algoritmos de Las Vegas

Exemplo: ORDENAÇÃO

Algoritmo: Quick Sort



Algoritmos de Las Vegas

Exemplo: ORDENAÇÃO

<u>Quick Sort</u>

- pivô escolhido deterministicamente
- pior caso: $O(n^2)$

<u>Quick Sort Randomizado</u>

- pivô escolhido aleatoriamente
- tempo esperado (para qualquer entrada!!):

?

Algoritmos de Las Vegas

Quick Sort Randomizado

Entrada: $a_1, a_2, a_3, \ldots, a_n$

Saída: $y_1, y_2, y_3, \ldots, y_n$

X = "número de comparações realizadas" = ?

$$X_{j,k} = \begin{cases} 1, \text{ se é feita a comparação entre } y_j \text{ e } y_k \\ 0, \text{ caso contrário} \end{cases}$$

Bernoulli

$$X = X_{1,2} + X_{1,3} + \ldots + X_{n-1,n}$$

Linearidade da Esperança:

$\mathbf{E}[f(X)] = f(\mathbf{E}[X])$

$$\mathbf{E}[X] = \mathbf{E}[X_{1,2} + X_{1,3} + \ldots + X_{n-1,n}] =$$

$$= \mathbf{E}[X_{1,2}] + \mathbf{E}[X_{1,3}] + \ldots + \mathbf{E}[X_{n-1,n}]$$

Algoritmos de Las Vegas

Quick Sort Randomizado

Entrada:

8	5	3	9	11	1	0	18	50	4	7

Saída:

0	1	3	4	5	7	8	9	11	18	50
			y_j				y_k			

Pr { "sucesso" } = **Pr** {"$X_{j,k}$= 1"} = $2 / (k - j + 1)$

E $[X] = \Sigma_{1 \leq j < k \leq n}$ **E** $[X_{j,k}]$ =

$= \Sigma_{1 \leq j < k \leq n} \; 2 / (k - j + 1) =$

$= O(n \log n)$

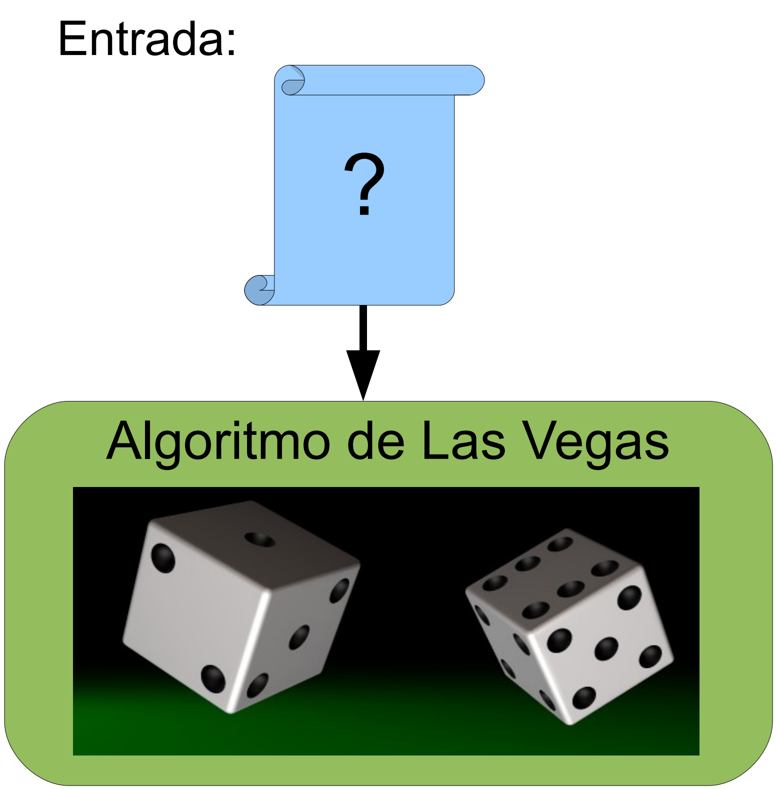
Algoritmos de Las Vegas

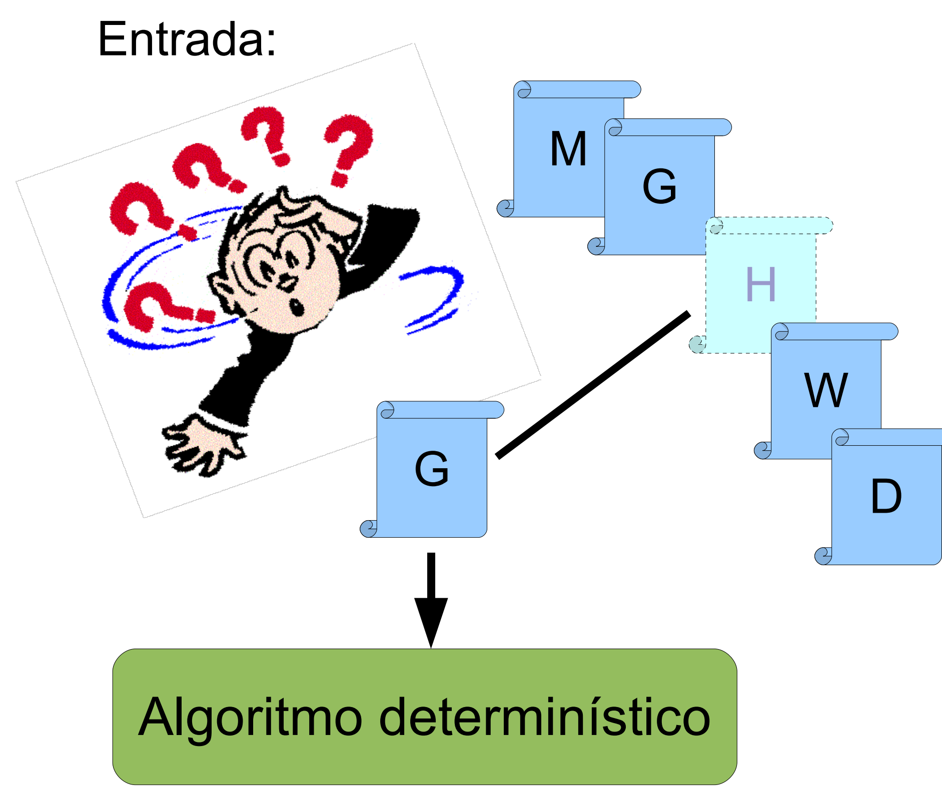
Tempo esperado de um algoritmo de Las Vegas

X

Tempo médio de um algoritmo determinístico

(dado um modelo probabilístico da entrada)





Monte Carlo X Las Vegas

- Monte Carlo,
  a partir de Las Vegas

1) enquanto tempo < t
2) se Las Vegas encontra SIM,
3) responda SIM
4) se Las Vegas encontra NÃO,
5) responda NÃO
6) reponda NÃO (arbitrariamente)

→ Monte Carlo baseado-no-SIM

→ X = "tempo do Las Vegas"

→ $\mathbf{Pr}$ {"erro"} $= \mathbf{Pr}\,\{C_S , A_N\}$
$= \mathbf{Pr}\,\{C_S\} \,.\, \mathbf{Pr}\,\{ A_N \mid C_S \}$
$\leq \mathbf{Pr}\,\{ \text{“}X \geq t\text{”} \}$

Markov! Chebyshev!

- Las Vegas,
  a partir de 2 Monte Carlos

1) repita
2) se MC-SIM encontra SIM,
3) responda SIM
4) se MC-NÃO encontra NÃO,
5) responda NÃO

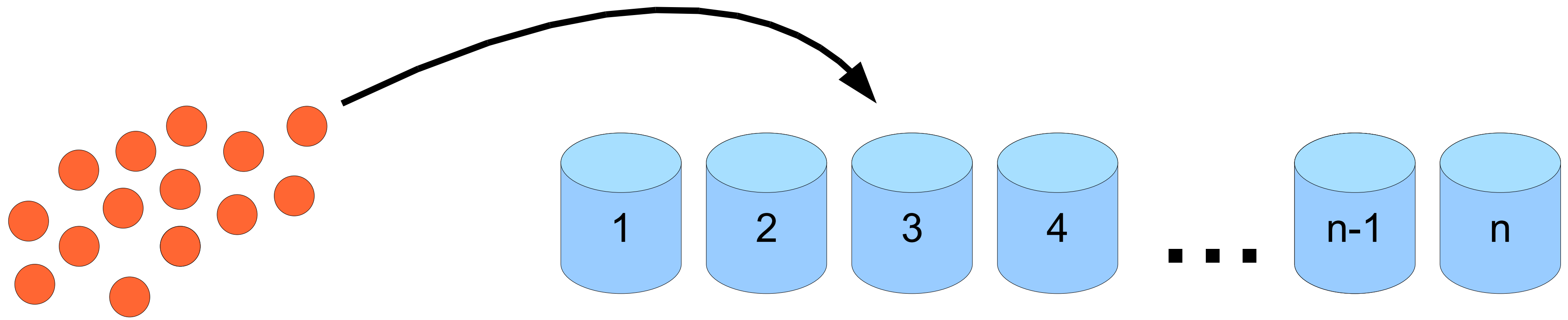
→ número T de iterações

V. A. geométrica!

→ Pr {"sucesso"} $= p = 1 - \boldsymbol{\varepsilon}_{\mathbf{SIM}} \,.\, \boldsymbol{\varepsilon}_{\mathbf{NÃO}}$

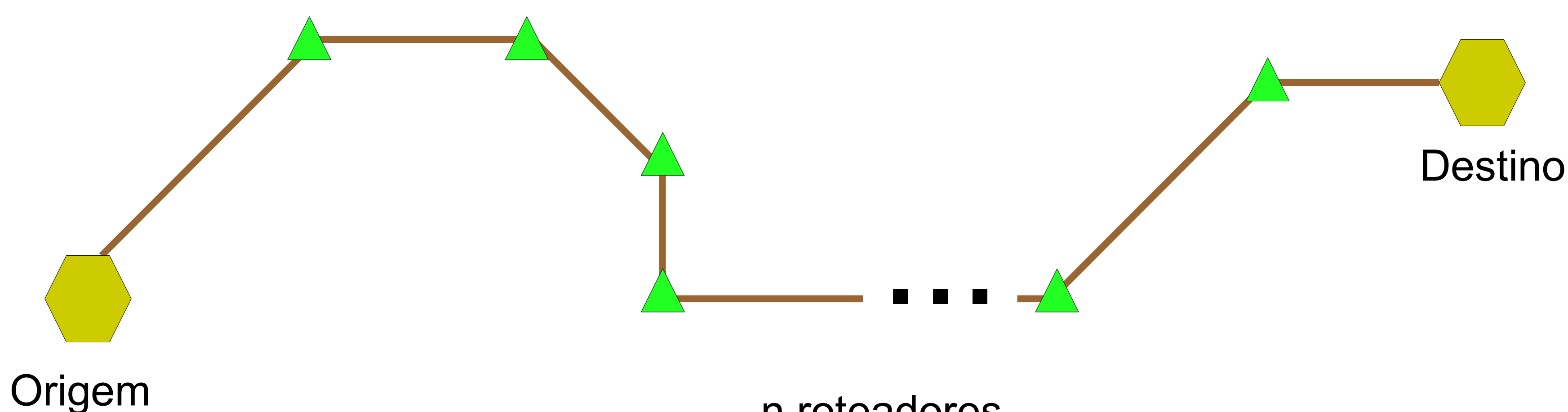
→ $\mathbf{E}\,[T] = 1 / p$

Modelo de bolas e latas



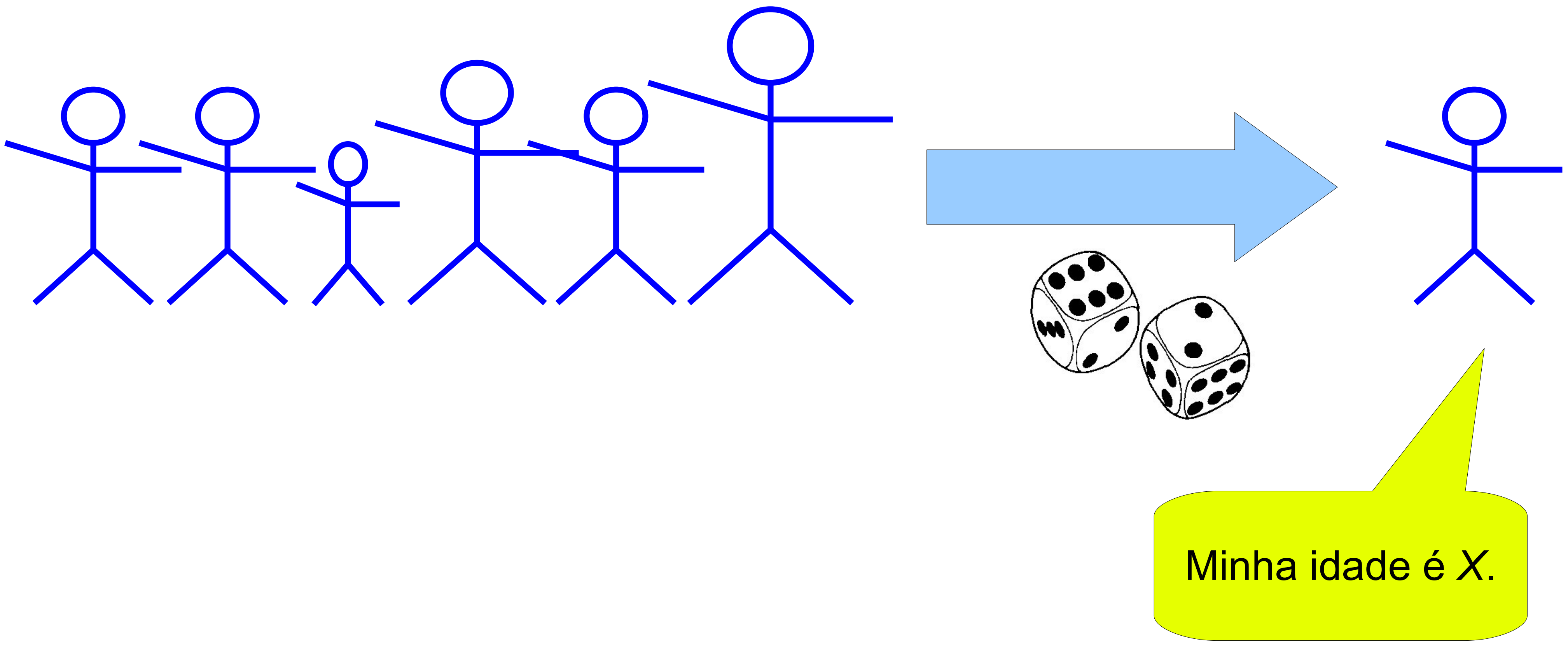
O colecionador de coupons

Exemplo: IDENTIFICAÇÃO DE ROTEADORES



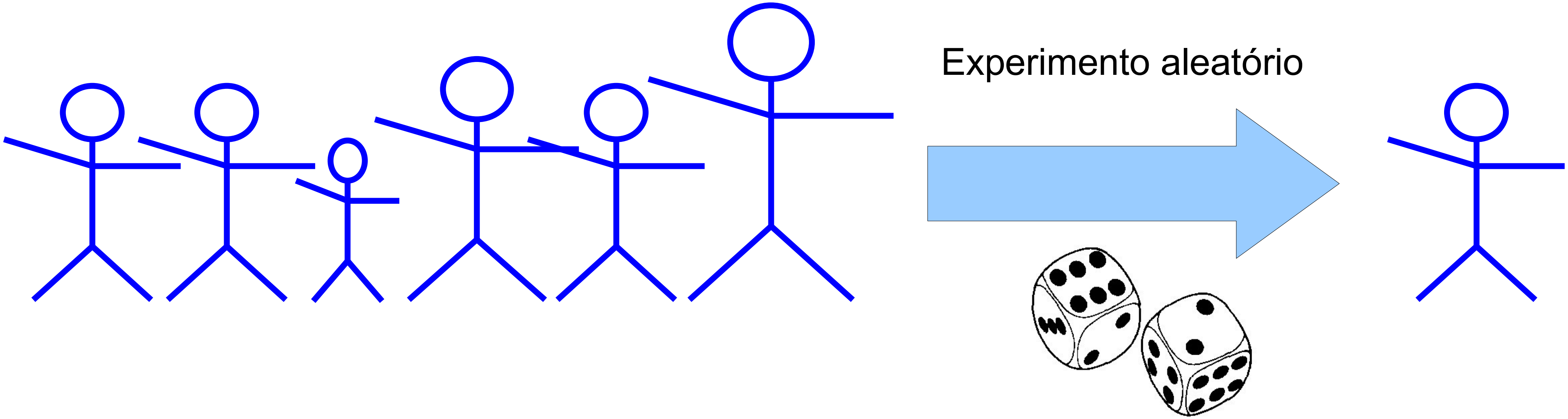
O Método Probabilístico

1) Método da esperança



O Método Probabilístico

1) Método da esperança

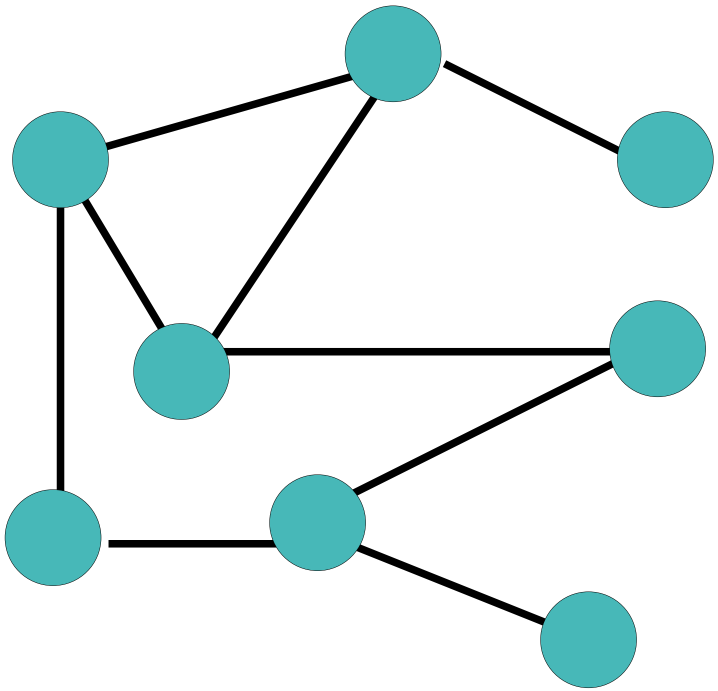


Se $\mathbf{E}[X] = \mu$, então
existe elemento para o qual $X \leq \mu$
e existe elemento para o qual $X \geq \mu$

O Método Probabilístico

1) Método da esperança

Exemplo: CORTES GRANDES EM GRAFOS

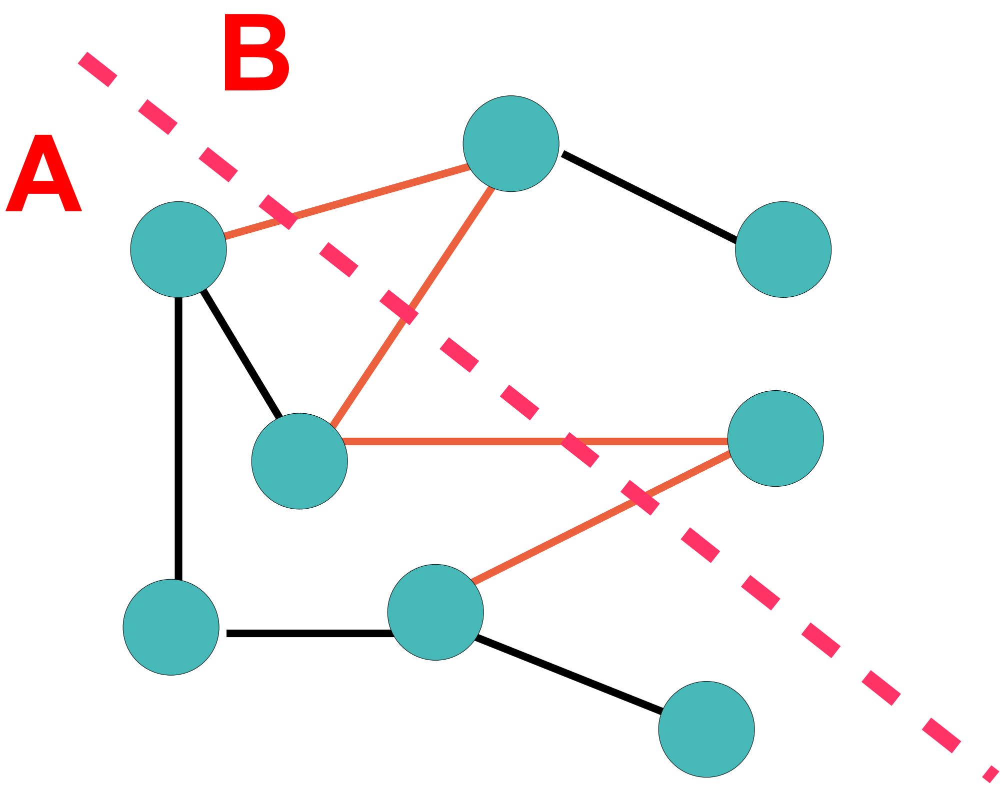


m arestas

O Método Probabilístico

1) Método da esperança

Exemplo: CORTES GRANDES EM GRAFOS



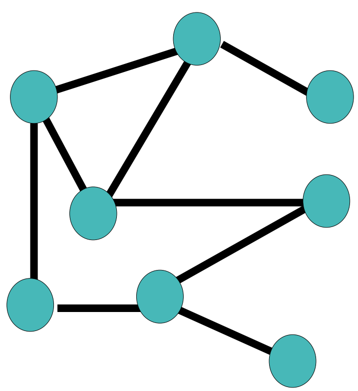
m arestas

Corte de tamanho 4

O Método Probabilístico

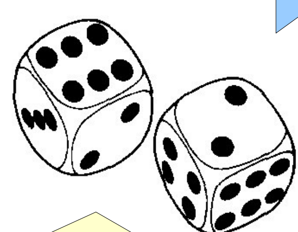
1) Método da esperança

Exemplo: CORTES GRANDES EM GRAFOS



m arestas

Algoritmo randomizado



Para cada vértice v...
    coloque v em A com probabilidade ½
    coloque v em B com probabilidade ½
Retorne o corte (A,B)

X = tamanho do corte retornado

$$X_j = \begin{cases} 1, & \text{aresta j pertence ao corte} \\ 0, & \text{caso contrário} \end{cases}$$

(Bernoulli)

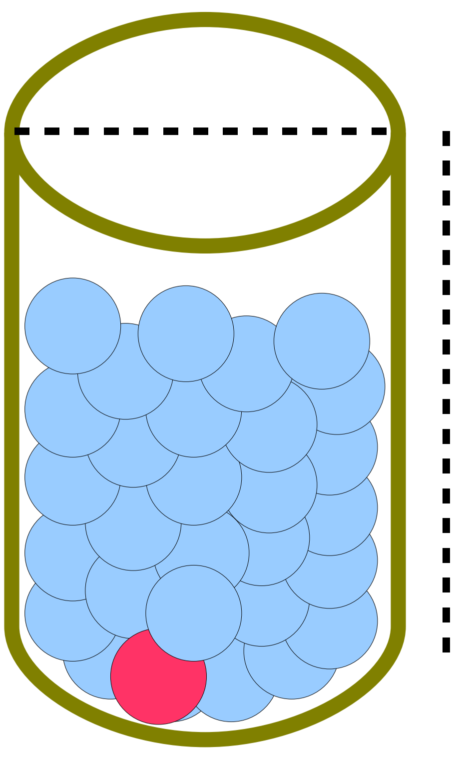
$\mathbf{Pr}$ {"sucesso"} = p = ?

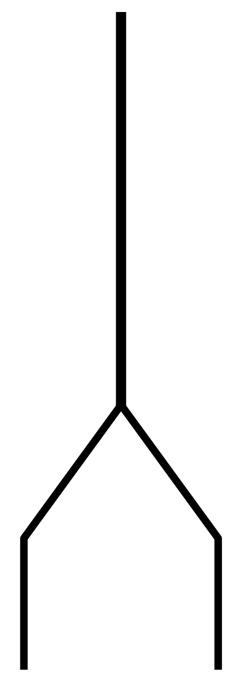
$$X = \Sigma_j X_j$$

$$\mathbf{E}[X] = \Sigma_j \mathbf{E}[X_j] = m \cdot p \quad = \quad m / 2$$

O Método Probabilístico

2) Método da probabilidade positiva

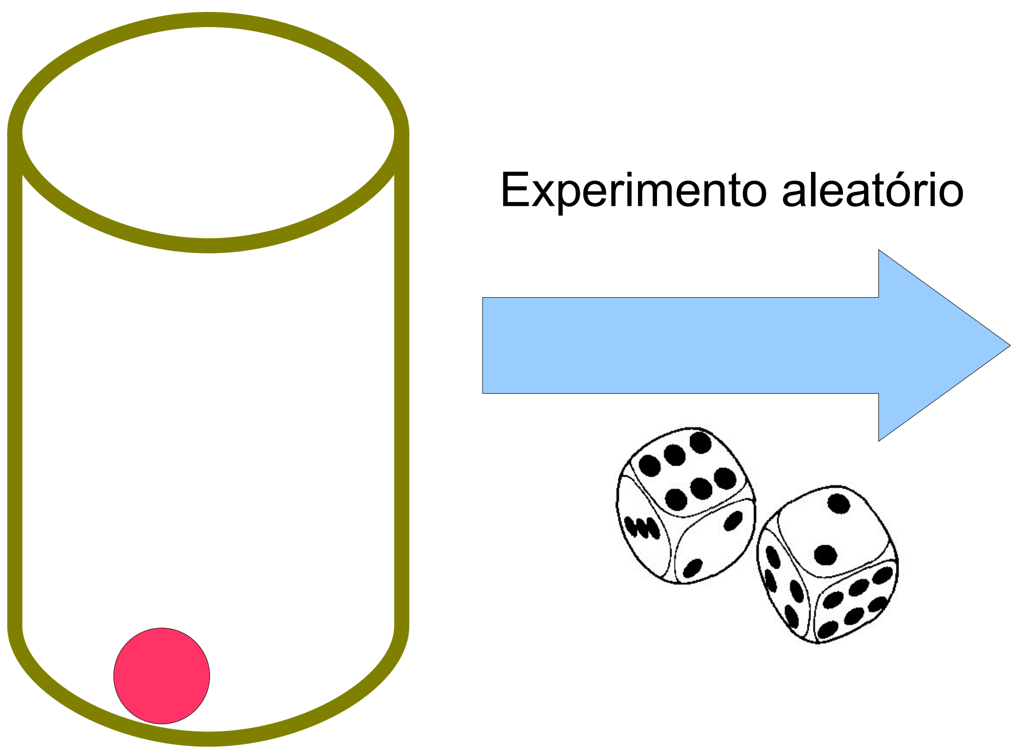




O Método Probabilístico

2) Método da probabilidade positiva

Espaço probabilístico Ω



Se

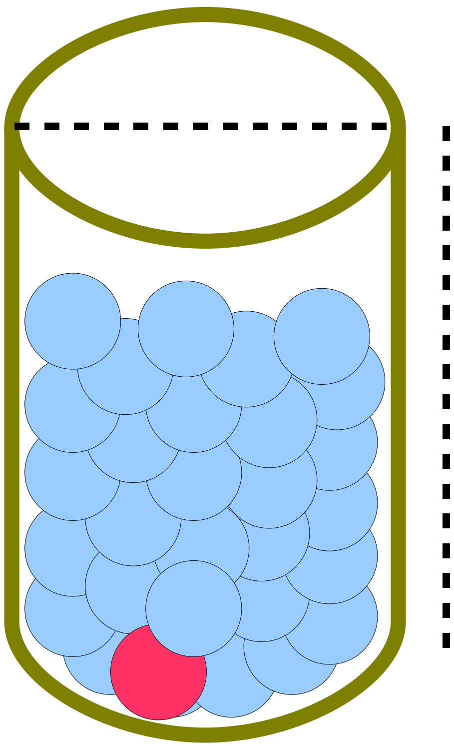
Pr {“ ”} > 0

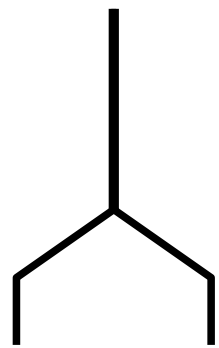
então

pertence a Ω

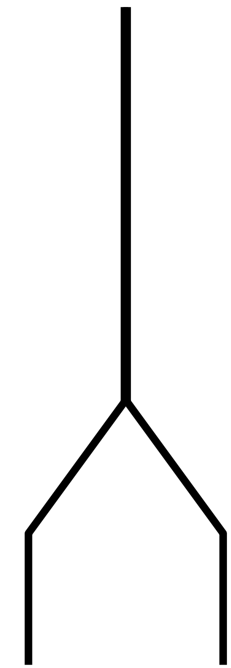
O Método Probabilístico

2) Método da probabilidade positiva





Algoritmo ruim
(não serve para a prova)



Algoritmo adequado
(serve para a prova)

Algoritmos Randomizados: Introdução

Celina Figueiredo
Guilherme Fonseca
Manoel Lemos
Vinícius Sá

impa

26º Colóquio Brasileiro de Matemática
IMPA – Rio de Janeiro – Brasil
2007