ANÁLISE PROBABILÍSTICA E ALGORITMOS RANDOMIZADOS

• O PROBLEMA DE CONTRATAÇÃO

- VAMOS ENTREVISTAR n CANDIDATOS
  - HÁ CANDIDATOS MELHORES QUE OUTROS, NÃO SABEMOS A ORDEM
  - DIGAMOS QUE O CANDIDATO c_i POSSUI UMA NOTA N_i

- SEMPRE TEMOS UM CANDIDATO CONTRATADO

- TODA VEZ QUE ENCONTRAMOS UM CANDIDATO MELHOR, TROCAMOS O CONTRATADO
  - NESTE CASO, PAGAMOS O CUSTO DE DEMISSÃO/CONTRATAÇÃO c_h
  - PODEMOS INCLUIR O CUSTO DE ENTREVISTAR TAMBÉM!

- GOSTARÍAMOS DE ESTIMAR O CUSTO AO FINAL DO PROCESSO

CONTRATAR-ASSISTENTE (A, n)

1. MELHOR = 0 — CANDIDATO DUMMY, PIOR QUE TODOS OS OUTROS
2. FOR $i = 1$ TO n
3. ENTREVISTA CANDIDATO i
4. SE $N_i >$ MELHOR
5. MELHOR = i
6. CONTRATA i

- CLARAMENTE O TEMPO DE EXECUÇÃO É $\Theta(n)$

- O CUSTO TOTAL É $c_h \cdot m$, ONDE m É O NÚMERO DE CONTRATAÇÕES FEITAS, I.E., O NÚMERO DE EXECUÇÕES DA LINHA 6.

- NO PIOR CASO TEMOS $N_1 < N_2 < \dots < N_n$, QUE POSSUI CUSTO EXATAMENTE $n \cdot c_h$.
  - QUAL A PROBABILIDADE DISSO OCORRER?

- NO MELHOR CASO: $N_1 > N_2 > \dots > N_n \Rightarrow c_h$

Análise Probabilística

- Em geral usamos para analizar o tempo de processamento
- Podemos usar para avaliar outras quantidades
  - ⇝ Como o custo total

- Para isso devemos ter alguma informação (ou fazer suposições) sobre a distribuição das instâncias.

  - ⇝ E aí fazemos uma média

  - ⇝ Chamamos de **tempo de processamento do caso médio** ou **custo do caso médio**

- No caso do prob. de contratação, podemos assumir que os candidatos vêm em uma **ordem aleatória**.
  - ⇝ Que as notas são uma ordenação de $1, \ldots, n$.
  - ⇝ E que cada uma das $n!$ possíveis ordenações possui igual probabilidade.

  - ⇝ Alternativamente dizemos permutação aleatória uniforme

- Podemos mudar o modelo do problema e supor que a agência de contratação nos manda uma lista de candidatos da qual sorteamos uma ordenação

```
Contratar-Assistente(A, n)
1 Randomiza A
2 Melhor = 0                    Candidato dummy, pior que todos os outros
3 for i = 1 to n
4     entrevista candidato i
5     se N_i > Melhor
6         Melhor = i
7         contrata i
```

- DIZEMOS QUE UM ALGORITMO É RANDOMIZADO SE SEU COMPORTAMENTO NÃO DEPENDE APENAS DA INSTÂNCIA, MAS TAMBÉM DE UM GERADOR DE NÚMEROS ALEATÓRIOS

  - VAMOS ASSUMIR A EXISTÊNCIA DE UMA FUNÇÃO RANDOM(a,b) QUE DEVOLVE UM NÚMERO EM $[a,b]$ DE FORMA UNIFORME.

    ⇝ CADA VALOR COM PROBABILIDADE $1/(b-a+1)$.

    ⇝ NA PRÁTICA É PSEUDO ALEATÓRIO

- O TEMPO DE PROCESSAMENTO ESPERADO É TOMADO SOBRE A DISTRIBUIÇÃO DO GERADOR DE NÚMEROS ALEATÓRIOS.

  ⇝ É DIFERENTE DA ENTRADA SER ALEATÓRIA.
  QUE CHAMAMOS DE TEMPO DE PROC. DO CASO MÉDIO

- ALGORITMOS RANDOMIZADOS "PAGAM" COM INCERTEZA (NO TEMPO OU NO CUSTO) E GANHAM EM SIMPLICIDADE, VELOCIDADE.

VARIÁVEIS ALEATÓRIAS INDICADORAS

- DADO UM ESPAÇO S E UM EVENTO A, A VARIÁVEL INDICADORA $I\{A\}$ ASSOCIADA AO EVENTO A É DEFINIDA POR

$$I\{A\} = \begin{cases} 1 & \text{SE } A \text{ ACONTECE} \\ 0 & \text{SE } A \text{ NÃO ACONTECE} \end{cases}$$

EX: $S = \{\overset{H}{\text{CARA}}, \overset{T}{\text{COROA}}\}$ COM $Pr\{H\} = Pr\{T\} = 1/2$.

↝ $I\{H\}$ CONTA O NÚMERO DE CARAS AO JOGAR A MOEDA UMA VEZ

↝ JOGANDO A MOEDA UMA VEZ, O NÚMERO ESPERADO DE CARAS É

$$E[I\{H\}] = 1 \cdot Pr\{H\} + 0 \cdot Pr\{T\} = 1/2$$

LEMA: DADO UM ESPAÇO S E UM EVENTO A EM S, TEMOS

$$E[I\{A\}] = Pr\{A\}$$

PROVA: $E[I\{A\}] = 1 \cdot Pr\{A\} + 0 \cdot Pr\{\bar{A}\} = Pr\{A\}$

↝ JOGANDO n MOEDAS IGUAIS, SEJA X_i A VARIÁVEL INDICADORA DO EVENTO EM QUE A i-ÉSIMA MOEDA DÁ CARA. SEJA X O NÚMERO DE CARAS. TEMOS

$$X = \sum X_i$$

$$\Rightarrow E[X] = E\left[\sum X_i\right] \overset{\text{LIN. DA ESPERANÇA}}{=} \sum E[X_i] = n/2$$

Análise do Problema de Contratação

- Seja X_i a variável indicadora de que o candidato i foi contratado, i.e., $X_i = I\{\text{o candidato } i \text{ foi contratado}\}$ e $X = X_1 + \dots + X_m$

- Pelo lema, temos $E[X_i] = Pr\{\text{o candidato } i \text{ foi contratado}\}$

↝ O i-ésimo candidato é contratado precisamente quando é melhor que os anteriores.

Af: $Pr\{\text{o candidato } i \text{ foi contratado}\} = 1/i$

Prova: Dado conjunto $S \subseteq \{1, \dots, n\}$ t.q. $|S| = i$, considere o conjunto P_S de todas as permutações de $\{1, \dots, n\}$ nas quais os primeiros i elts são os elts em S.

Obs: $|P_S| = i! \cdot (n-i)!$

Há $(i-1)!\,(n-i)!$ permutações em P_S onde i é contratado. i.e., em $1/i$ das permutações de P_S, i é contratado. Logo

$$Pr\{\text{o candidato } i \text{ foi contratado}\} = \frac{\#\text{ Permutações em que } i \text{ é cont.}}{n!}$$

$$= \frac{1}{n!} \cdot \sum_{|S|=i} \frac{1}{i} \cdot |P_S| = \frac{1}{i} \cdot \frac{1}{n!} \cdot \sum_i |P_S| = \frac{1}{i} \cdot \frac{1}{n!} \cdot n! = 1/i$$

Note que $\{P_S : |S| = i\}$ é uma partição do conj. de permutações. Logo, $\sum_i |P_S| = n!$

□

- Finalmente, $E[X] = \sum E[X_i] = \sum_{i=1}^{n} 1/i \leq \ln n + O(1)$

- Ou seja, por mais que entrevistemos n candidatos, contratamos, em média, $\ln n$ deles.

⇒ Custo médio $O(c_h \cdot \ln n)$ — Bem melhor que o pior caso!

Recapitulando

- O primeiro algoritmo assumia que a ordem dos candidatos era aleatória, e, em média, tem um custo de $O(C_h \ln m)$.

  ⇝ Um adversário poderia criar uma instância cujo custo é exatamente $C_h \cdot m$.

- O segundo algoritmo realiza um "sorteio" de forma a aleatorizar a ordem dos candidatos dentro de si próprio, e tem um custo esperado de $O(C_h \ln m)$.

  ⇝ Nenhum adversário pode criar uma instância que aumente este custo

- Essa é a diferença entre análise probabilística e algoritmos randomizados

COMO PRODUZIR UMA ORDEM ALEATÓRIA USANDO UM GERADOR DE NÚMEROS?

↝ OU COMO PERMUTAR UM ARRAY DE FORMA A OBTER UMA ORDENAÇÃO ALEATÓRIA UNIFORME.

↝ ATRIBUIR PRIORIDADES AOS ELEMENTOS DO ARRAY

EX: $A = (1,2,3,4)$, $P = (36,3,62,19) \Rightarrow B = (2,4,1,3)$

PERMUTAR-COM-ORDENAÇÃO (A)

1. $n = A.\text{LENGTH}$
2. $P[1..n] \leftarrow$ NOVO ARRAY
3. FOR $i = 1$ TO n
4. $\quad P[i] = \text{RANDOM}(1, n^3)$
5. ORDENAR A DE ACORDO COM P.

$O(n \log n)$

↝ POR QUE ESCOLHER NÚMEROS ALEATÓRIOS ENTRE 1 E n^3?

↝ É POSSÍVEL PROVAR QUE A PROBABILIDADE DE TODAS AS PRIORIDADES SEJAM DIFERENTES É $1 - 1/n$.

PROVA: $\dfrac{\#\text{ NÃO REPETEM}}{\text{TOTAIS}} = \dfrac{n^3(n^3-1)\cdots(n^3-(n-1))}{n^{3n}} = \prod_{i=0}^{n-1}\left(1 - \dfrac{i}{n^3}\right)$

AF: $\forall i, \prod_{i=0}^{k}\left(1 - \dfrac{i}{n^3}\right) \geq 1 - \dfrac{k}{n^2}$

INDUÇÃO: $\prod_{i=0}^{k+1}\left(1 - \dfrac{i}{n^3}\right) = \left(1 - \dfrac{k+1}{n^3}\right)\prod_{i=0}^{k}\left(1 - \dfrac{i}{n^3}\right) \geq \left(1 - \dfrac{k+1}{n^3}\right)\left(1 - \dfrac{k}{n^2}\right)$

$$= 1 - \frac{k+1}{n^3} - \frac{k}{n^2} + \frac{k(k+1)}{n^5}$$

NOTE QUE, COMO $\dfrac{k+1}{n^3} \leq \dfrac{n}{n^3}$, ENTÃO $\dfrac{k+1}{n^3} + \dfrac{k}{n^2} \leq \dfrac{k+1}{n^2}$

LOGO

$$1 - \frac{k+1}{n^3} - \frac{k}{n^2} + \frac{k(k+1)}{n^5} \geq 1 - \frac{k+1}{n^2}.$$

↝ PRECISAMOS MOSTRAR QUE A DISTRIBUIÇÃO É UNIFORME.

PROBABILIDADE

- ESPAÇO AMOSTRAL Ω É O CONJUNTO DE TODOS OS POSSÍVEIS RESULTADOS DE UM EXPERIMENTO ALEATÓRIO.

  EX: FACES DE UM DADO $\Omega = \{1, \dots, 6\}$

- UM EVENTO É QUALQUER SUBCONJUNTO DE Ω

  EX: FACES PARES $A = \{2, 4, 6\}$

- UM EVENTO ELEMENTAR É UM EVENTO DE TAMANHO 1, $A = \{r\}$

  EX: "SAIU 3" $A = \{3\}$

- UMA FUNÇÃO DE PROBABILIDADE É UMA FUNÇÃO $Pr: \Omega \to [0,1]$ TAL QUE $\sum_{w \in \Omega} Pr(w) = 1$ (NO CASO DISCRETO, FINITO)

  ⇝ ESTENDEMOS Pr PARA SUBCONJUNTOS DE Ω DE TAL FORMA QUE

  SE $A \subseteq \Omega$, ENTÃO $Pr(A) = \sum_{x \in A} Pr(x)$.

  LOGO: 1. PARA TODO $A \subseteq \Omega$, TEMOS $0 \leq Pr(A) \leq 1$

  2. $Pr(\Omega) = 1$

  3. SE $A_1, \dots, A_k$ SÃO EVENTOS DOIS A DOIS DISJUNTOS, ENTÃO

  $$Pr\left(\bigcup_{i=1}^{k} A_i\right) = \sum_{i=1}^{k} Pr(A_i)$$

Princípio da inclusão-exclusão:

4. Se $A_1, \dots, A_k$ são eventos distintos, então

$$Pr\left(\bigcup A_i\right) = \sum Pr(A_i) - \sum_{i<j} Pr(A_i \cap A_j) + \sum_{i<j<k} Pr(A_i \cap A_j \cap A_k)$$
$$- \dots + (-1)^{\ell+1} \sum_{i_1 < \dots < i_\ell} Pr\left(A_{i_1} \cap \dots \cap A_{i_\ell}\right) + \dots$$

Union Bound

- Uma versão "frouxa" do Princ. da Inc.-Exc.

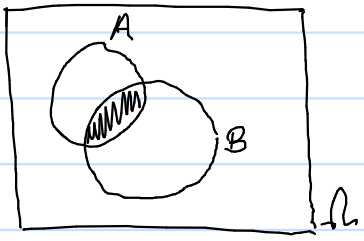
5. Se $A_1, \dots, A_k$ são eventos distintos, então

$$Pr\left(\bigcup A_i\right) \leq \sum Pr(A_i)$$

Probabilidade Condicional

- A probabilidade de ocorrer um evento A, sabendo que ocorre um evento B

$$Pr(A \mid B) = \frac{Pr(A \cap B)}{Pr(B)}$$



Ex: Dados honestos ($Pr(x) = 1/6 \ \forall x$).
Qual a probabilidade de sair 2 sabendo que foi par?

$$Pr(2 \mid \text{foi par}) = \frac{Pr(\{2\})}{Pr(\{2,4,6\})} = \frac{1/6}{1/2} = 1/3$$

- Se $Pr(A \mid B) = Pr(A)$, ou seja, $Pr(A \cap B) = Pr(A) \cdot Pr(B)$, dizemos que A e B são independentes

→ Saber que o resultado do exp. é B, não altera a prob. de que seja A.

Probabilidade Total

- Se particionamos Ω em eventos (disjuntos) $B_1, \dots, B_m$, a prob. de um evento A é

$$Pr(A) = \sum_{i=1}^{m} Pr(A \mid B_i) \cdot Pr(B_i)$$

- Daí sai a regra de Bayes

$$Pr(B_k \mid A) = \frac{Pr(B_k \cap A)}{Pr(A)} = \frac{Pr(A \mid B_k) \cdot Pr(B_k)}{\sum_{i=1}^{m} Pr(A \mid B_i) \cdot Pr(B_i)}$$

Variáveis aleatórias (VA)

- Uma **variável aleatória** é uma função $X : \Omega \to \mathbb{R}$.

  Ex: tempo de execução ou custo de um Alg. randomizado

- Quando escrevemos $\Pr(X = x)$, nos referimos à **probabilidade do evento**

$$A_{X=x} = \{w \in \Omega : X(w) = x\}$$

Ex: jogar n moedas e contar o número de caras. $\Pr(H) = \Pr(T)$

Se X é o número de caras (e a moeda é honesta)

$$\Pr(X = k) = \binom{n}{k} / 2^k$$

- A **função densidade de probabilidade** de X é a função

$$P_X : \mathbb{R} \to [0,1] \quad \text{t.q.} \quad P_X(x) = \Pr(X = x)$$

- A **esperança** ou **valor esperado** de uma variável aleatória X é a média ponderada (pela probabilidade) de seus possíveis valores

$$E[X] = \sum_x x \cdot P_X(x)$$

Ex: lançar seis moedas. $E[X] = 1 \cdot P_X(1) + \cdots + 6 \cdot P_X(6)$

onde $P_X(k) = \binom{6}{k} / 2^6$ $= (6 + 30 + 60 + 60 + 30 + 6)/2^6 = 3$

→ A esperança é uma **função linear**: se X, Y são VA, então

$$E[aX + bY] = a\,E[X] + b\,E[Y]$$

Ex: lançamento de seis moedas, $X = X_1 + \cdots + X_6$, onde X_i é o resultado da i-ésima moeda:

$$E[X] = \sum E(X_i) = 6 \cdot \tfrac{1}{2} = 3$$

Limites de Cauda

- A esperança é a média

→ Valioso quando repetimos um experimento várias vezes.

→ Não nos dá muita informação quanto à dens. de prob.

No caso de não sabermos a densidade

Desigualdade de Markov:

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

Prova: $E[X] = \sum_x x \cdot P(X \geq x) \geq \sum_{x \geq a} x \cdot Pr(X \geq x) \geq \sum_{x \geq} a \cdot Pr(X \geq x)$

$$= a \sum_{x \geq a} Pr(X \geq x) = a \cdot Pr(X \geq a)$$ ∎

- A variância de X é a esperança da distância da média

$$Var[X] = E[(X - E[X])^2 = E[X^2] - E[X]^2$$

Desigualdade de Chebychev:

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

Prova: $Pr(|X - E[X]| \geq a) = Pr((X - E[X])^2 \geq a^2)$

$$\overset{\text{Markov}}{\leq} \frac{E[(X - E[X])^2]}{a^2} = \frac{Var[X]}{a^2}$$

Alternativamente: $Pr(|X - E[X]| \geq k \cdot \sigma) \leq 1/k^2$, onde $\sigma = \sqrt{Var[X]}$

Variáveis aleatórias importantes

Bernoulli

- Comumente usada como variável aleatória indicadora

$$I_A = \begin{cases} 1 & \text{se } A \text{ ocorreu} \\ 0 & \text{se } A \text{ não ocorreu} \end{cases}$$

$$P_{I_A}(x) = \begin{cases} 1-p & \text{se } x=0 \\ p & \text{se } x=1 \\ 0 & \text{para todos valores de } x \neq 0,1. \end{cases}$$

Ex: Moeda viciada: $P_R(H) = p$, $P_R(T) = 1 - P_R(H) = 1-p$

Prop: $E[I_A] = 0 \cdot P_{I_A}(0) + 1 \cdot P_{I_A}(1) = p$

Binomial

- Indica o número de sucessos em uma seq. de exp. aleat. idênticos e independentes

$$X = X_1 + \dots + X_m \quad \text{com} \quad P_{X_1}(x) = \dots = P_{X_m}(x) \ \forall x$$

Ex: Número de caras

- Abreviamos usando o número m de indicadoras e a probabilidade p de sucesso: $B(m,p)$

- Densidade $P_X(x) = \binom{m}{x} p^x (1-p)^{m-x}$, $0 \leq x \leq m$ inteiro.

$P_X(x) = 0$ caso contrário

Variável aleatória Geométrica

- Quando ao invés de interessados no número de sucessos em uma seq. de experimentos, estamos interessados no número de repetições até o primeiro sucesso.

ex: Quantas vezes precisamos lançar uma moeda para que saia cara

- A densidade de uma geométrica X com prob. de sucesso p é

$$P_X(x) = \begin{cases} p(1-p)^{x-1} & \text{para } x = 1, 2, 3, \dots \\ 0 & \text{caso contrário} \end{cases}$$

- A esperança de X é $E[X] = 1/p$.

Prova:

$$E[X] = \sum_{x=1}^{\infty} x\, p(1-p)^{x-1} \qquad (1)$$

$$(1-p)E[X] = \sum_{x=1}^{\infty} x\, p(1-p)^{x} = \sum_{x=2}^{\infty} (x-1)\, p(1-p)^{x-1} \qquad (2)$$

$$(1) - (2): \quad E[X] - (1-p)E[X] = p + \sum_{x=2}^{\infty} p(1-p)^{x-1} \cdot (x - (x-1))$$

$$= p + p \sum_{x=1}^{\infty} (1-p)^{x}$$

$$= p + p \cdot \frac{(1-p)}{p} = 1$$

$$\Rightarrow \quad pE[X] = 1$$

∎

Monte Carlo e Las Vegas

- Dois tipos de algoritmos que diferem no local de sua incerteza

  - Monte Carlo : na corretude — pode dar soluções erradas - mas gasta tempo determinístico
  - Las Vegas : no tempo de execução — apenas dá soluções corretas mas gasta tempo aleatório

Monte Carlo

- Decisão

  ~> Erro unilateral:

  1. Baseados no SIM : nunca erram quando dizem SIM
  2. Baseados no NÃO : nunca erram quando dizem NÃO

  ~> Erro bilateral : podem estar errados em ambas as respostas.

Ex: Identidade de polinômios : Dados $F(x) = (x-a_1)(x-a_2)\dots(x-a_d)$
e $G(x) = b_d x^d + b_{d-1}x^{d-1} + \dots + b_0$
decidir se $F = G$.

~> Podemos expandir F e comparar os coeficientes encontrados: $O(d^2)$
~> Testar as raízes de F em G : $O(d^2)$

Ou sortear um inteiro r entre 1 e $100d$, testar se $F(r)=G(r)$
se $F(r) \neq G(r)$, retornamos NÃO, caso contrário retornamos SIM.
↳ certificado de NÃO

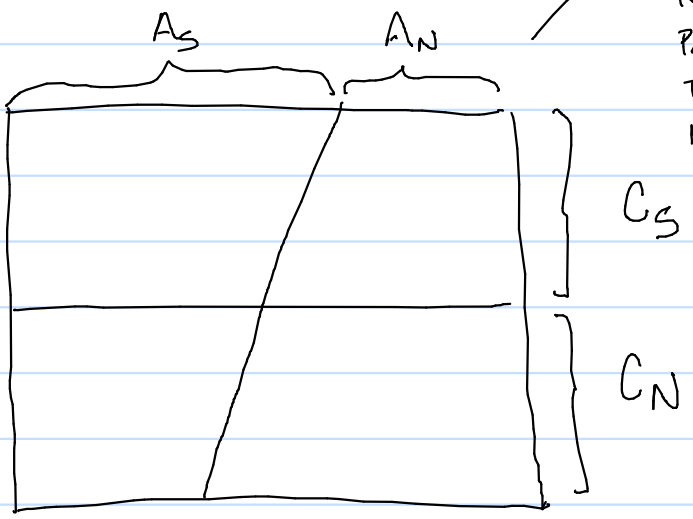
O polinômio $F(x) - G(x)$ tem no máximo d raízes em $[1, 100d]$
A probabilidade de r ser uma dessas raízes é $d/100d = 1/100$.

Custo? $O(d)$

Probabilidades Monte Carlo Unilateral

- Considere os seguintes eventos

1. A_S - o algoritmo responde SIM
2. A_N - o algoritmo responde NÃO
3. C_S - a resposta correta é SIM
4. C_N - a resposta correta é NÃO



Não é bem assim porque o algoritmo pode dar respostas diferentes para a mesma instância

- Gostaríamos de estudar as probabilidades

$$Pr(A_S \mid C_S),\ Pr(A_N \mid C_S),\ Pr(A_S \mid C_N),\ Pr(A_N \mid C_N)$$

- Se o alg. é baseado no SIM, então $A_S \subseteq C_S$.
  Ou seja $Pr(C_S \mid A_S) = 1$ e $Pr(C_N \mid A_S) = 0$

- Se o alg. é baseado no NÃO, então $A_N \subseteq C_N$
  Ou seja $Pr(C_N \mid A_N) = 1$ e $Pr(C_S \mid A_N) = 0$

Probs diferentes

Ex: Identidade de polinômios. O alg proposto apresenta um certificado quando responde NÃO. Logo $A_N \subseteq C_N$. É baseado no NÃO.

- Um algoritmo baseado no NÃO (resp. SIM) responde corretamente SIM (resp. NÃO) para instâncias em C_S (resp. C_N), i.e, $C_S \subseteq A_S$ ($C_N \subseteq A_N$) pois não há um certificado de NÃO (SIM)

- DEVEMOS NOS PREOCUPAR COM A PROB. DE ENCONTRAR UM CERTIFICADO.

- CONSIDERE UM ALGORITMO A BASEADO NO NÃO E QUE ERRA COM PROB NO MÁXIMO ε_1
  ↓ PARA INSTÂNCIAS EM C_N

  ⇝ ISSO É $Pr(A_S \mid C_N) \leq \varepsilon_1$

  - EXECUTE-O t VEZES ⇝ ATÉ QUE NÃO SEJA ENCONTRADA OU RESPONDA SIM ⇝ A^t

  - QUAL A PROB. GLOBAL ε_t DE ERRO DO NOVO ALGORITMO?

    ⇝ CERTAMENTE É UMA INSTÂNCIA EM C_N $A^1, \dots, A^t$
    ⇝ PARA A QUAL OBTIVEMOS SIM EM TODAS AS EXECUÇÕES DE A

$$\text{LOGO, } \varepsilon_t = Pr\left(C_N \wedge \bigwedge_{i=1}^{t} A_S^i\right) = Pr(C_N) \cdot Pr\left(\bigwedge A_S^i \mid C_N\right)$$

$$= Pr(C_N) \cdot Pr\left(A^t \mid C_N\right) \cdot Pr\left(\bigwedge_{i=1}^{t-1} A_i \mid C_N, A^t\right)$$

$$= Pr(C_N) \cdot \prod_{K=1}^{t} Pr\left(A_K \mid C_N \wedge \bigwedge_{J=1}^{K-1} A_i\right)$$

PORQUE OS A_S^i SÃO INDEP.

$$\overset{\downarrow}{=} Pr(C_N) \prod Pr\left(A_S^i \mid C_N\right)$$

$$= \underbrace{Pr(C_N)}_{\leq 1} Pr(A_S \mid C_N)^t \leq \varepsilon_1^t$$

Las Vegas

- O tempo computacional é uma variável aleatória
- Julgamos pela esperança, variância, ...

Ex: Busca em lista com repetição, queremos encontrar o 9.

↝ lista com n entradas
↝ números de 0 a 9
↝ cada número aparece exatamente $1/10$ das vezes.

- Se varremos a lista do começo pro final : $\Theta(n)$
  final pro começo : $\Theta(n)$
- ímpares depois pares : $\Theta(n)$

• Se escolhemos aleatoriamente uma posição, verificamos, e repetimos esta operação até encontrar o 9

↝ distribuição geométrica com prob. de sucesso $1/10$.
Logo, a esperança é $10 = O(1)$.

↝ Quer busca certificado? SIM

- Transformar Las Vegas em Monte Carlo : rodar o Las Vegas um número limitado de vezes, se não encontrar certificado, resolve não

→ A prob. de erro do MC = Pr do LV precisar de mais tempo
→ Seja X a VA de tempo do LV. Se o tempo máximo é $k\mu$ onde $\mu = E[X]$, temos

$$\varepsilon_{MC} \leq Pr[X \geq k\mu] \leq \frac{E[X]}{k\mu} = 1/k.$$

- TRANSFORMAR MC EM LV

⇝ MENOS EFICAZ. O ALGORITMO PODE RODAR PARA SEMPRE

EX: SE A É BASEADO NO NÃO, E REPETIMOS A ENQUANTO NÃO ENCONTRAR NÃO. O QUE ACONTE COM UMA INSTÂNCIA SI.

⇝ EM ALGUNS CASOS PODEMOS FAZER COM QUE AS EXECUÇÕES DO MC NÃO SEJAM INDEPENDENTES.
→ O ALG. PARARIA DEPOIS DE FAZER TODAS (OU ALGUMAS) ESCOLHAS POSSÍVEIS. ISSO SERIA UM CERTIFICADO.

⇝ SE HÁ DOIS ALGORITMOS, Alg_S E Alg_N BASEADOS EM SIM E NÃO, RESP. COM PROBS. P_S E P_N DE APRESENTAR CERT. P/ SIM E NÃO PODEMOS EXECUTAR OS DOIS EM CADA ITERAÇÃO, ATÉ QUE UM APRESENTE CERTIFICADO.

- A PROB. DE SUCESSO DE CADA EXEC É $\geq p = \text{MIN}\{P_S, P_N\}$

- LOGO, O # DE EXEC. ESPERADO É $1/p$.

EX: ENCONTRAR ESTRUTURA QUE SABEMOS QUE EXISTE
⇝ CORTE COM PELO MENOS $e(G)/c$ ARESTAS.

Classes de Complexidade

- **RP** (Randomized Polynomial time): Probs para os quais existem algoritmos que respondem SIM com prob $\geq \frac{1}{2}$ se $I \in C_S$, e NÃO com prob. 1 se $I \in C_N$

  ⇝ Probs. para os quais existe Alg de Monte Carlo baseado no SIM.

- **Co-RP**: Probs. t.q. $\exists$ Alg Monte Carlo baseado no NÃO.

- **ZPP**: (Zero-error Probabilistic Polinomial time)
  Probs. para os quais existe Alg. de Las Vegas de tempo esperado polinomial.

- **BPP** (Bounded-error Probabilistic Polinomial time)
  Probs. para os quais existe Alg de Monte Carlo bilateral onde tanto a prob de resp. SIM com $I \in C_S$, quanto responder NÃO com $I \in C_N$ são $\geq 3/4$.

- **PP** (Probabilistic Polinomial time)
  Probs. para os quais existe Alg de Monte Carlo bilateral onde tanto a prob de resp. SIM com $I \in C_S$, quanto responder NÃO com $I \in C_N$ são $> 1/2$.

Obs: $BPP \subseteq PP$. A diferença é o número de repetições do Alg para que o erro seja $< \varepsilon$.

⇝ BPP = Polinomial
PP = Exponencial

Paradigmas Combinatórios

Bolas e Latas

- Há m bolas idênticas
- Há n latas

- As bolas são associadas às latas aleatoriamente (com prob $1/n$)
- Uma lata pode receber mais de uma bola

- Quantas latas ficam vazias?
- Qual o número esperado de latas com mais do que k bolas?
- Quantas bolas precisamos p/ que existam mais latas com bolas do que vazias?
- Quantas bolas há na lata mais cheia.

Ex: n pessoas. Há duas pessoas que fazem aniversário no mesmo dia?

Seja A_i a probabilidade da bola i cair numa lata desocupada. A probabilidade de termos duas bolas na mesma lata é

$$p = \Pr\left(\bigwedge A_i\right) = \prod_{i=1}^{m} \Pr\left(A_i \,\middle|\, \bigwedge_{j<i} A_j\right)$$

Note que $\Pr\left(A_i \,\middle|\, \bigwedge_{j<i} A_j\right)$ é a probabilidade de i cair em uma lata, dado que há $i-1$ latas ocupadas, ou seja $\frac{n-i+1}{n}$

Logo

$$p = \prod_{i=1}^{m} \left(\frac{n-i+1}{n}\right)$$

Qual o valor de n para que seja mais provável haver duas pessoas com mesmo aniversário?

↝ Devemos resolver a equação $p < \frac{1}{2}$: $m \approx \sqrt{2n \ln n} = O(\ln n)$

Colecionador de Cupons

- Há um conjunto de cupons $D = \{d_1, \dots, d_n\}$
- Gostaríamos de ter um de cada

↝ Seq. de var. ind $X_1, \dots$ cada uma indica o índice que saiu.
↝ Cada índice tem prob. $1/n$ de sair.

Defina $W_{n,k}$ = # de sorteios realizados até que se tenha obtido k itens distintos, dos n existentes

↝ Estamos interessados em $E[W_{n,n}]$.

↝ Parece com bolas x latas
- sorteios ↝ itens existentes
- n latas, infinitas bolas

Defina Z_i = quantidade de sorteios para que o # de cupons distintos vá de $i-1$ p/ i.

↝ Quando o colecionador possui $i-1$ cupons distintos a prob. de sorteamos um cupom diferente é $p_i = 1 - \frac{i-1}{n}$

↝ Logo, cada Z_i é uma V.A. geométrica com prob. de sucesso p_i

$\Rightarrow W_{n,k} = Z_1 + \dots + Z_k$ e

$$E[W_{n,k}] = \sum_{i=1}^{k} \frac{n}{n-i+1} \Rightarrow E[W_{n,n}] = \sum_{i=1}^{n} \frac{n}{n-i+1} = n \sum_{i=1}^{n} \frac{1}{i}$$

$$= n \ln n + \Theta(n).$$

EX: MENSAGEM EM PACOTES CLIENTE → SERVIDOR

- CAMINHOS LONGOS (MUITOS ROTEADORES)
- UM PACOTE PODE GUARDAR A INFO DE NO MAX UM ROTEADOR
  ⇝ OU OCUPARIA MUITO ESPAÇO / BANDA
- SE HÁ n ROTEADORES, CADA PACOTE GUARDA A INFO DE UM ROTEADOR COM PROB $1/n$.

⇒ O NÚMERO DE PACOTES QUE PRECISAM SER RECEBIDOS PARA TERMOS INFO DE TODOS OS ROTEADORES É

$$n \log n + \Theta(n)$$

COMO FAZER PARA QUE CADA ROTEADOR APAREÇA COM PROB. $1/n$.

- AS VEZES O PACOTE NÃO SABE QUANTOS ROTEADORES TEM NO CAM.

ESTRATÉGIA:

1) GUARDO INFO DO PRIMEIRO ROTEADOR
2) NO k-ÉSIMO ROTEADOR, TROCA A INFO GUARDADA PELA INFO DO ROTEADOR ATUAL COM PROB $1/k$.

→ PARA QUE A INFO DO k-ÉSIMO ROTEADOR CHEGUE NO SERVIDOR, PRECISAMOS TROCAR NO PASSO 2 ($\text{PROB. } 1/k$) E NÃO TROCAR DEPOIS ($\text{PROB.} \prod_{i=k+1}^{n} \frac{i-1}{i}$)

⇒ $\Pr(\text{GUARDAR A } k\text{-INFO}) = \frac{1}{k} \cdot \frac{k}{k+1} \cdot \frac{k+1}{k+2} \cdot \ldots \cdot \frac{n-1}{n} = \frac{1}{n}$

Análise Probabilística

- Algoritmos determinísticos executam da mesma forma para a mesma instância. Por isso os analizamos pelo pior caso.

- Na prática, pode ser que o caso típico seja muito melhor que o pior caso. Ou seja, que a probabilidade de ocorrer o pior caso seja muito baixa.

→ Analizar o alg. levando em consideração a distribuição das instâncias é chamado de análise probabilística.

- De algs determinísticos e randomizados!

Ex: Quicksort. Pior caso: $O(n^2)$
Caso médio: $O(n \log n)$

Obs: Podemos randomizar o quicksort!

Quicksort-randomizado (S)
1. reordene S de forma aleatória
2. Quicksort (S)

- Diferentes execuções de algs. randomizados possuem tempos diferentes p/ a mesma instância. Por isso analizamos o caso médio.

BUCKET SORT

- RODA EM TEMPO LINEAR!
- QUAL O MODELO PARA ISSO?

  → UMA ENTRADA COM $n = 2^m$ NÚMEROS
  → ESCOLHIDO DE FORMA ALEATÓRIA E UNIFORME NO INTERVALO $[0, 2^k[$ COM $k > m$.

- HÁ n BUCKETS ROTULADOS DE 0 A $n-1$ EM BINÁRIO
- CADA NÚMERO É COLOCADO NO BUCKET QUE REPRESENTA SEUS PRIMEIROS m DÍGITOS BINÁRIOS.

  → FEITO EM TEMPO $O(n)$

- SEJA X_j O # DE ELTS NO BUCKET j.

  → BOLAS = ENTRADA, LATAS = BUCKETS } X_j É V.A. BINOM. $B(n, 1/n)$

- ORDENAMOS CADA BUCKET INDIVIDUALMENTE (USANDO, POR EX, O QUICKSORT

  → TEMPO QUADRÁTICO EM X_j : $c X_j^2$

- O VALOR ESPERADO PARA O TEMPO TOTAL $X = \sum_j c X_j^2$ É

$$E[X] = E\left[\sum_j c X_j^2\right] = c \cdot n \cdot E[X_1^2].$$

  → LEMBREMOS QUE COMO X_1 É V.A. BIN. $B(n, p)$, TEMOS

$$E[X_1] = n(n-1)p^2 + np$$

  → NESTE CASO, $p = 1/n$ E, PORTANTO, $E[X_1] = 2 - 1/n < 2$.
  ⇒ BUCKET SORT RODA EM TEMPO ESPERADO $O(n)$

O MÉTODO PROBABILÍSTICO

- PROVAS DE EXISTÊNCIA

- É POSSÍVEL DE-RANDOMIZAR ALGS. RANDOMIZADOS.

EX: É POSSÍVEL COLORIR AS ARESTAS DO K_{40} COM DUAS CORES SEM CLIQUE MONOCROMÁTICA DE TAMANHO 8?

- HÁ 780 ARESTAS. LOGO 2^{780} COLORAÇÕES

↝ SEJAM $C_1, \dots, C_{\binom{40}{8}}$ AS CLIQUES DE TAMANHO 8 EM $G = K_{40}$.

↝ SEJA M_i O EVENTO "TODAS AS ARESTAS DE C_i SÃO AZUIS OU TODAS AS ARESTAS DE C_i SÃO VERMELHAS"

↝ $Pr(M_i) = \frac{1}{2^{\binom{8}{2}-1}} = 2^{-27}$

QUEREMOS CALCULAR $Pr(\bigcap \overline{M_i}) = 1 - Pr(\bigcup M_i)$

PELO LIMITE DA UNIÃO, TEMOS

$$Pr(\bigcup M_i) \leq \sum Pr(M_i) = \frac{\binom{40}{8}}{2^{27}} < 0.573$$

LOGO, $Pr(\bigcap \overline{M_i}) > 0.427 > 0$ ∎

O MÉTODO DA ESPERANÇA

- MAXCUT : ENCONTRAR UMA BIPARTIÇÃO (A,B) DE G QUE MAXIMIZE $e(A,B)$.

- NP-DIFÍCIL

TEOREMA : EXISTE CORTE DE TAMANHO $\geq e(G)/2$.

ENTRADA (G)
SAÍDA UM CORTE DE G

CORTE (G)
1. CRIE DOIS CONJS. A E B
2. P/ CADA VÉRTICE, COLOQUE EM A COM PROB $1/2$, SENÃO COLOQUE EM B
3. RETORNE (A,B)

- A PROB. DE UMA ARESTA ESTAR NO CORTE É $1/2$.
- O NÚMERO ESPERADO DE ARESTAS É $e(G) \cdot \frac{1}{2} = e(G)/2$

- É POSSÍVEL DE-RANDOMIZAR ESTE ALGORITMO

→ EM CADA PASSO, EM VEZ DE ESCOLHER ENTRE A E B ALEATORIAMENTE, ESCOLHEMOS A PARTIÇÃO QUE MAXIMIZA O NÚMERO DE ARESTAS NO CORTE.