Melhores momentos

AULA PASSADA

Problemas polinomiais

Analise de um algoritmo em um determinado modelo de computação estima o seu consumo de tempo e quantidade de espaço como uma função do tamanho da instância do problema.

Exemplo: o consumo de tempo do algoritmo EUCLIDES (a, b) é expresso como uma função de $\langle a \rangle + \langle b \rangle$.

Um problema é solúvel em tempo polinomial se existe um algoritmo que consome tempo $\mathrm{O}(\langle I \rangle^c)$ para resolver o problema, onde c é uma constante e I é instância do problema.

Exemplos

- Máximo divisor comum

  Tamanho da intância: $\lg a + \lg b$

  Consumo de tempo Café-Com-Leite é $O(b)$ (não-polinomial)

  Consumo de tempo EUCLIDES é $O(\lg b)$ (polinomial)

- Subseqüência comum máxima

  Tamanho da instância: $n + m$
  Consumo de tempo REC-LCS-LENGTH é $\Omega(2^{\min\{m,n\}})$ (não-polinomial)

  Consumo de tempo LCS-LENGTH é $\Theta(mn)$ (polinomial).

Mais exemplos

- Problema booleano da mochila
  Tamanho da instância: $\lg n + n \lg W + n \lg V + \lg W$
  Consumo de tempo MOCHILA-BOOLEANA é $\Theta(nW)$
  (não-polinomial)
- Problema fracionário da mochila
  Tamanho da instância: $\lg n + n \lg W + n \lg V + \lg W$
  Consumo de tempo MOCHILA-FRACIONÁRIA é
  $\Theta(n \lg n)$
  (polinomial).
- Ordenação de inteiros $A[1\,.\,.\,n]$
  Tamanho da instância: $n \lg M$,
  $M := \max\{|A[1]|, |A[2]|, \ldots, |A[n]|\} + 1$
  Consumo de tempo MERGE-SORT é $\Theta(n \lg n)$
  (polinomial).

Classe P

Por um algoritmo eficiente entendemos um algoritmo polinomial.

A classe de todos os problemas de decisão que podem ser resolvidos por algoritmos polinomiais é denotada por P (classe de complexidade).

Exemplo: As versões de decisão dos problemas:

> máximo divisor comum, subseqüência comum máxima e mochila fracionária

estão em P.

Para muitos problemas, não se conhece algoritmo essencialmente melhor que "testar todas as possibilidades". Em geral, isso não está em P.

AULA 23

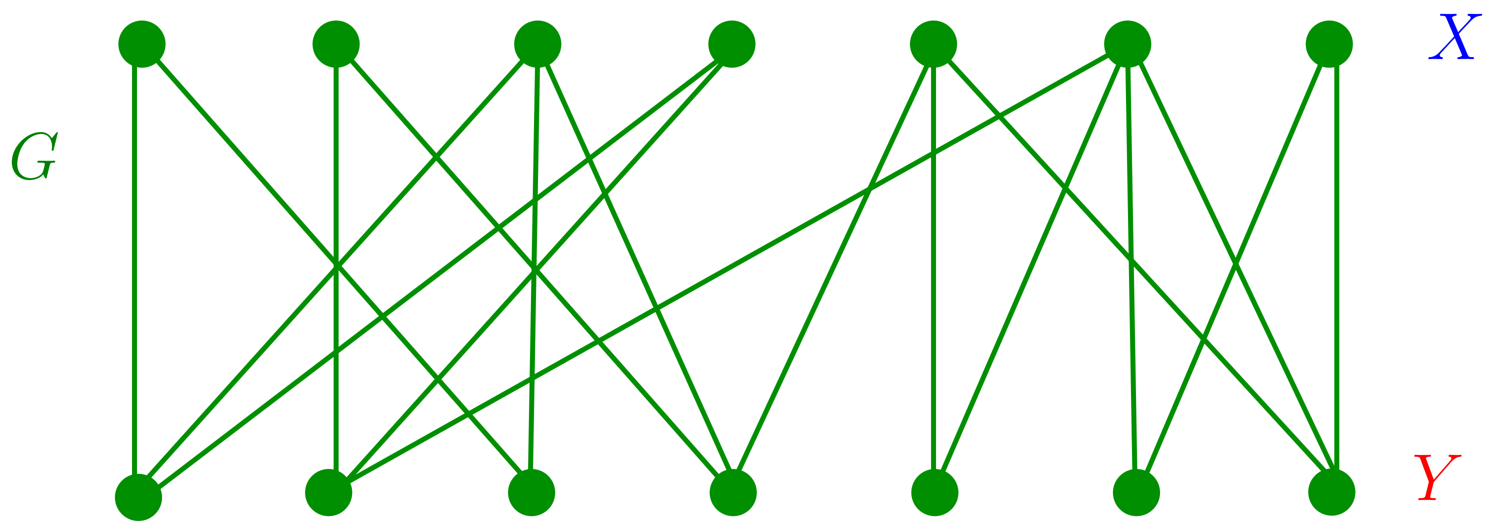
Mais complexidade computacional

CLR 36 ou CLRS 34

Arthur e Merlin

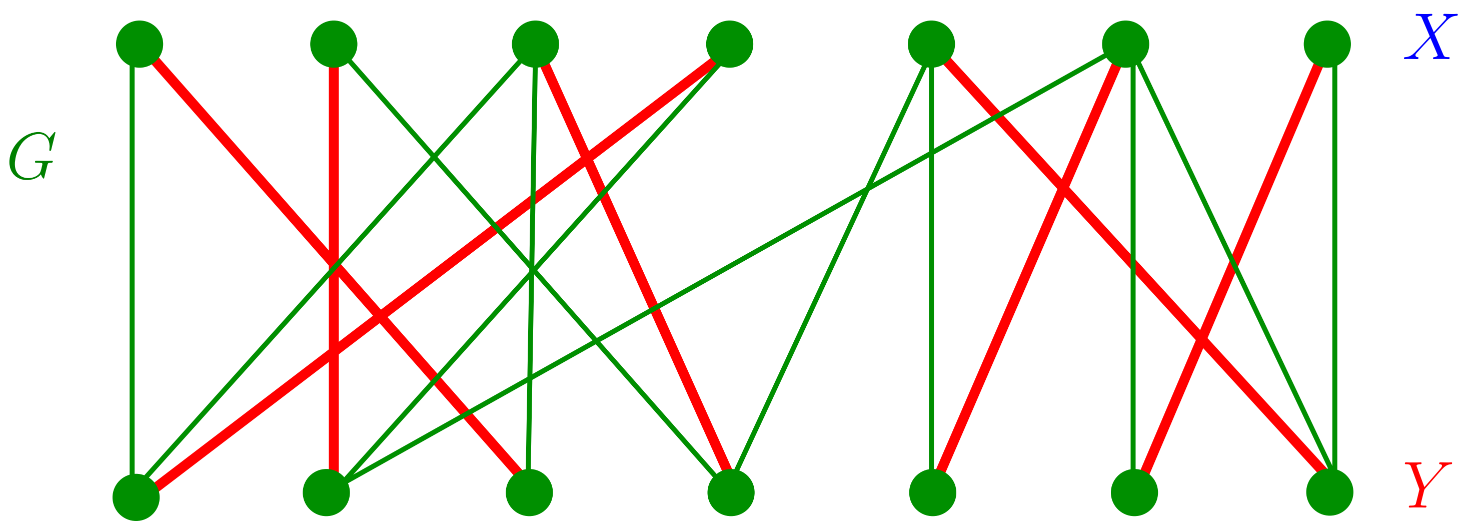
Emparelhamentos

Problema: Dado um grafo bipartido encontrar um emparelhamento perfeito.



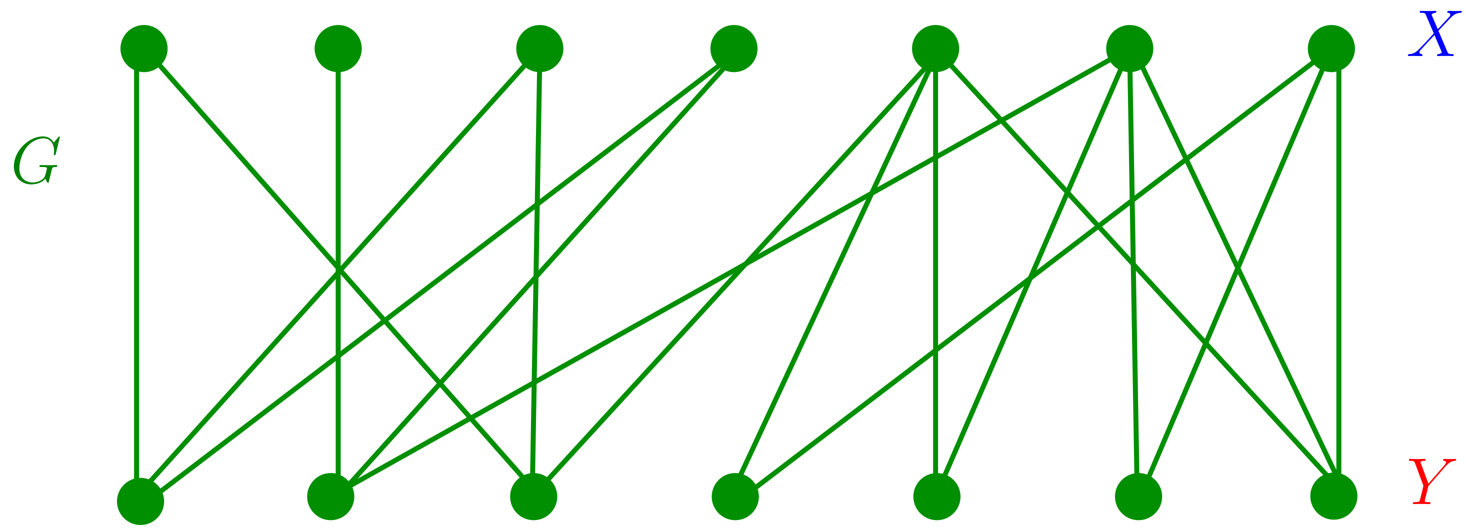
Emparelhamentos

Problema: Dado um grafo bipartido encontrar um emparelhamento perfeito.



Emparelhamentos

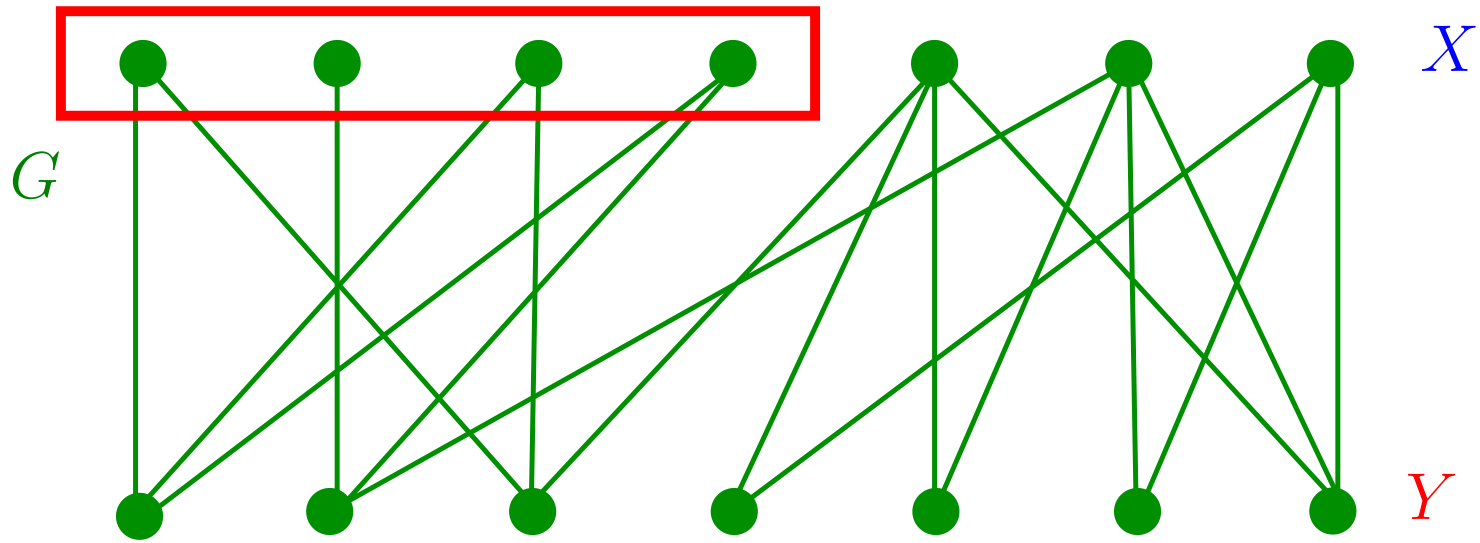
Problema: Dado um grafo bipartido encontrar um emparelhamento perfeito.



NÃO existe! Certificado?

Emparelhamentos

Problema: Dado um grafo bipartido encontrar um emparelhamento bipartido.

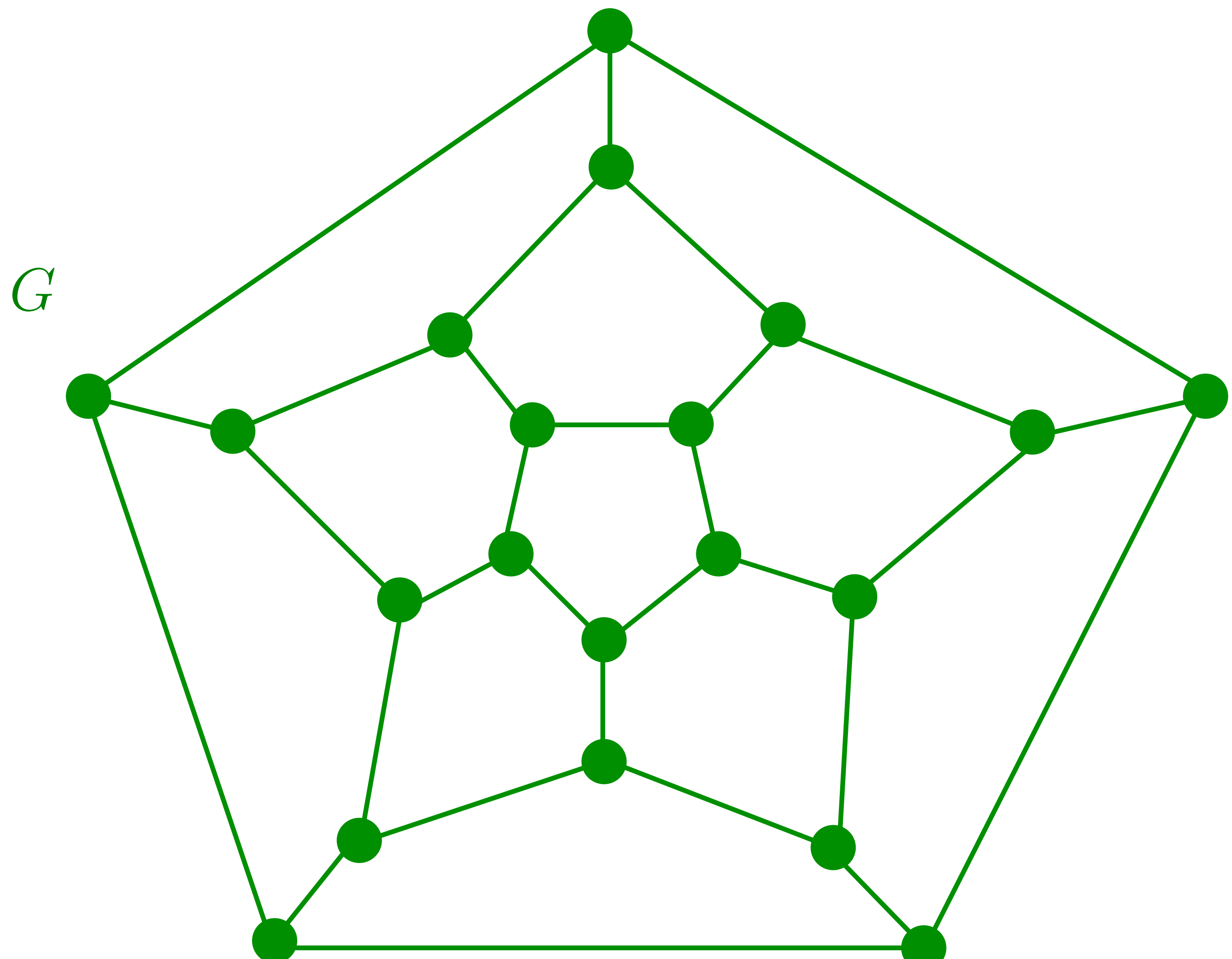


NÃO existe! Certificado: $S \subseteq X$ tal que $|S| > |\text{vizinhos}(S)|$.
Teorema de Hall: G tem um emparelhamento perfeito se e somente se

$$|S| \leq |\text{vizinhos}(S)|, \quad \text{para todo } S \subseteq X.$$

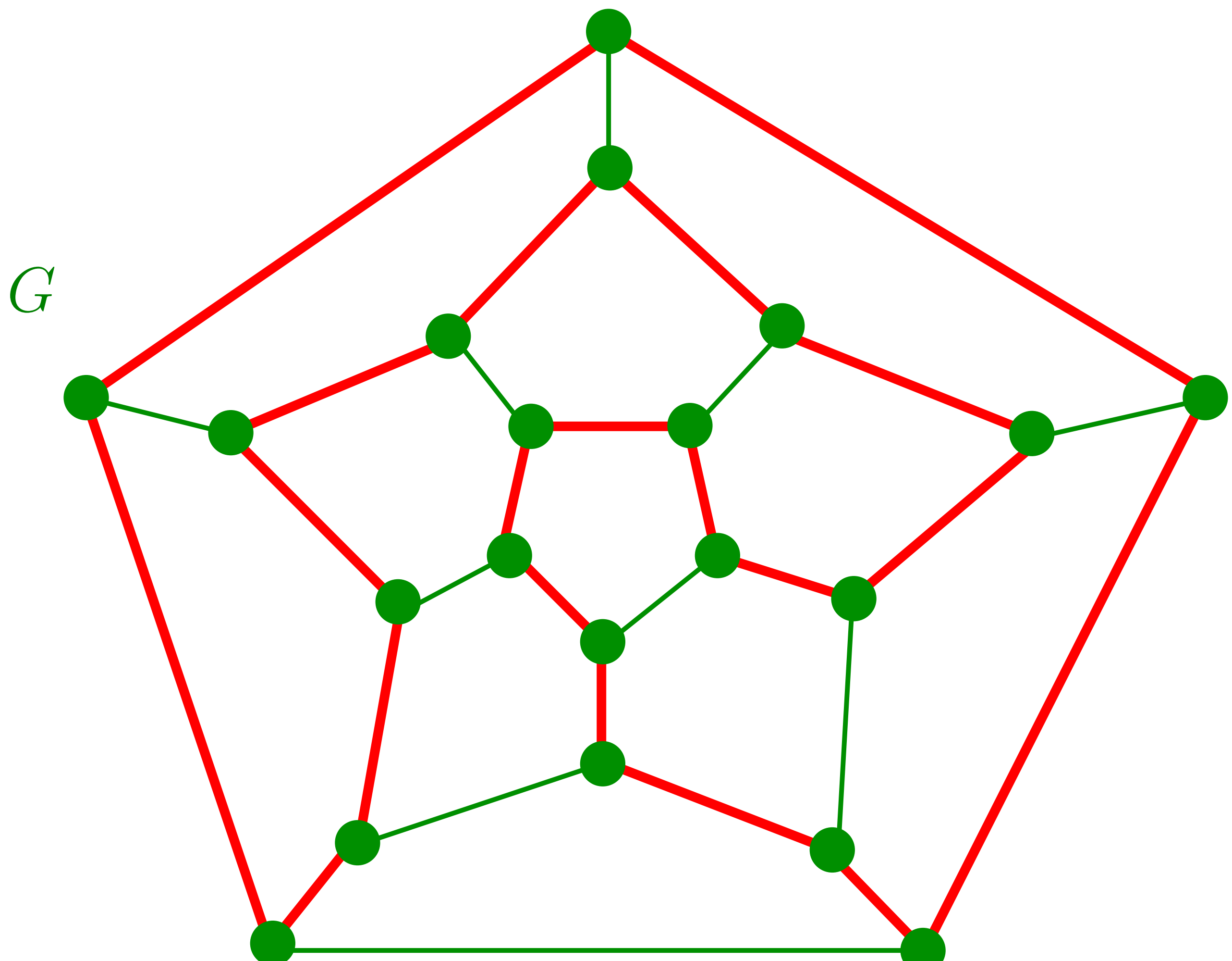
Grafos hamiltonianos

Problema: Dado um grafo encontrar um ciclo hamiltoniano.



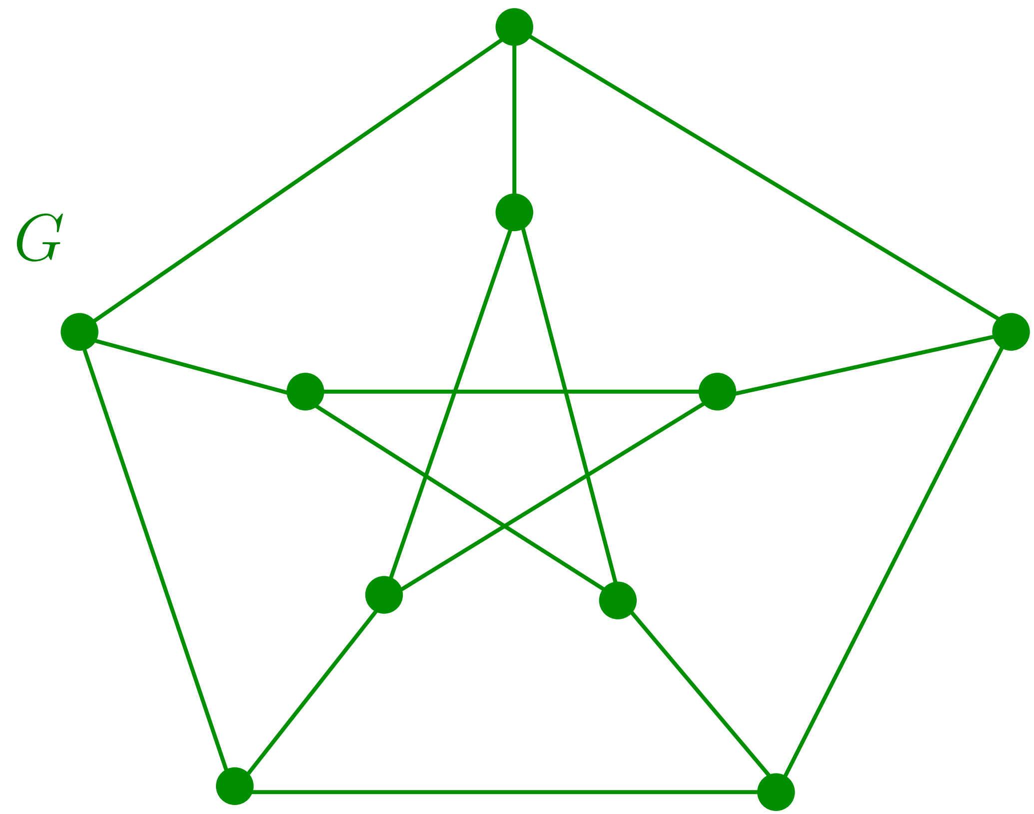
Grafos hamiltonianos

Problema: Dado um grafo encontrar um ciclo hamiltoniano.



Grafos hamiltonianos

Problema: Dado um grafo encontrar um ciclo hamiltoniano.



NÃO existe! Certificado? Hmmm . . .

Verificador polinomial para SIM

Um verificador polinomial para a resposta SIM de um problema Π é uma algoritmo polinomial ALG(I, C) que

recebe uma instância I de Π e um objeto C, $\langle C\rangle = \mathrm{O}(p(\langle I\rangle))$,

e

devolve SIM se e somente se $\Pi(I) =$ SIM,

onde $p(n)$ é um polinômio.

O objeto C é dito um certificado polinomial ou certificado curto da resposta SIM a $\Pi(I)$.

Exemplos

- Se G é hamiltoniano, então um ciclo hamiltoniano de G é um certificado polinomial:

  dados um grafo G e C pode-se verificar em tempo $\mathrm{O}(\langle G \rangle)$ se C é um ciclo hamiltoniano.

- se $X[1\,..\,m]$ e $Y[1\,..\,n]$ possuem uma ssco $\geq k$, então uma subseqüência comum $Z[1\,..\,k]$ é um certificado polinomial resposta:

  dados $X[1\,..\,m], Y[1\,..\,n]$ e $Z[1\,..\,k]$ pode-se verificar em tempo $\mathrm{O}(m+n)$ se Z é ssco de X e Y.

- se n é um número composto, então um divisor $d > 1$ de n é um certificado polinomial.

Classe NP

Formada pelos problemas de decisão que possuem um verificador polinomial para a resposta SIM.

Em outras palavras, um problemas de decisão Π está em NP se existe um problema Π' em P e uma função polinomial $p(n)$ tais que, para cada instância I de Π, existe um objeto C, $\langle C \rangle = \mathrm{O}(p(\langle I \rangle))$, e

$$\Pi(I) = \text{SIM se e somente se } \Pi'(I, C) = \text{SIM}.$$

O objeto C é dito um certificado polinomial ou certificado curto da resposta SIM de $\Pi(I)$.

Exemplos

Problemas de decisão com certificado polinomial para SIM:

- existe subseqüência crescente $\geq k$?
- existe subcoleção disjunta $\geq k$ de intervalos?
- existe mochila booleana de valor $\geq k$?
- existe mochila de valor $\geq k$?
- existe subseqüência comum $\geq k$?
- grafo tem ciclo de comprimento $\geq k$?
- grafo tem ciclo hamiltoniano?
- grafo tem emparelhamento (casamento) perfeito?

Todos esses problemas estão em NP.

$\mathbf{P} \subseteq \mathbf{NP}$

Prova:

se Π é um problema em P, então pode-se tomar a seqüência de instruções realizadas por um algoritmo polinomial para resolver $\Pi(I)$ como certificado polinomial da resposta SIM a $\Pi(I)$.

Outra prova:

Pode-se construir um verificador polinomial para a resposta SIM a Π utilizando-se um algoritmo polinomial para Π como subrotina e ignorando-se o certificado C.

Exemplo de certificado

Seqüência de operações do algoritmo EUCLIDES:

```
mdc(317811,514229)
  mdc(514229,317811)
    mdc(317811,196418)
      mdc(196418,121393)
        mdc(121393,75025)
          mdc(75025,46368)
            mdc(46368,28657)
              mdc(28657,17711)
                mdc(17711,10946)
                  mdc(10946,6765)
                    mdc(6765,4181)
                      mdc(4181,2584)
                        mdc(2584,1597)
                          mdc(1597,987)
                            mdc(987,610)
                              mdc(610,377)
```

Exemplo de certificado (cont.)

```
mdc(377,233)
   mdc(233,144)
      mdc(144,89)
         mdc(89,55)
            mdc(55,34)
               mdc(34,21)
                  mdc(21,13)
                     mdc(13,8)
                        mdc(8,5)
                           mdc(5,3)
                              mdc(3,2)
                                 mdc(2,1)
                                    mdc(1,0)
```

certificado polinomial para mdc$(317811, 514229) = 1$

Outro certificado

EXTENDED- EUCLIDES (a, b) devolve d junto com x, y tais que $ax + by = d$

- podemos verificar se $d \mid a$ e $d \mid b$
- podemos verificar se $ax + by = d$

Se $d' \mid a$ e $d' \mid b$ então

$$d' \mid (ax + by) = d$$

e portanto $d' \leq d$

Conclusão: $d = \text{mdc}(a, b)$ e x, y são um certificado curto deste fato.

$\mathbf{P} \neq \mathbf{NP}$?

É crença de muitos que a classe NP é maior que a classe P, ainda que isso

não tenha sido provado até agora.

Este é o intrigante problema matemático conhecido pelo rótulo "P $\neq$ NP?"

Não confunda NP com "não-polinomial".

Verificador polinomial para NÃO

Um verificador polinomial para a resposta NÃO de um problema Π é uma algoritmo polinomial ALG(I, C) que

> recebe uma instância I de Π e um objeto C,
> $\langle C \rangle = \mathrm{O}(p(\langle I \rangle))$,

e

> devolve SIM se e somente se $\Pi(I) =$ NÃO,

onde $p(n)$ é um polinômio.

O objeto C é dito um certificado polinomial ou certificado curto da resposta NÃO a $\Pi(I)$.

Exemplos

- Se um grafo G com bipartição X e Y não possui um emparelhamento perfeito então $S \subseteq X$ tal que $|S| > |\text{vizinhos}(S)|$ é um certificado polinomial desse fato:

  dados um grafo G com bipartição X e Y e $S \subseteq X$ pode-se verificar em tempo $\mathrm{O}(\langle G \rangle)$ se $|S| > |\text{vizinhos}(S)|$.

- se n não é um número primo, então um divisor $d > 1$ de n é um certificado polinomial desse fato.

Classe co-NP

A classe co-NP é definida trocando-se SIM por NÃO na definição de NP.

Um problema de decisão Π está em co-NP se admite um certificado polinomial para a resposta NÃO.

Os problemas em NP $\cap$ co-NP admitem certificados polinomiais para as respostas SIM e NÃO.

Em particular, P $\subseteq$ NP $\cap$ co-NP.

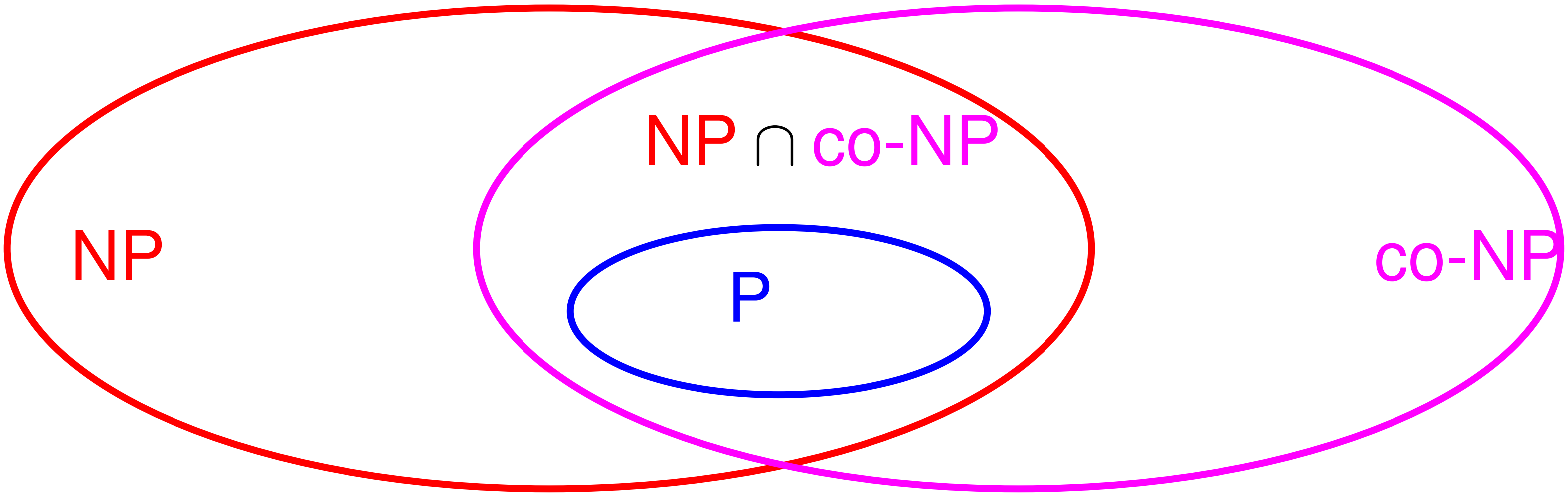
Exemplos

Problemas de decisão com certificado polinomial para NÃO:

- não existe subseqüência crescente $\geq k$?
- um dado número é primo?
- não existe subseqüência comum $\geq k$?
- grafo não tem emparelhamento (casamento) perfeito?

Todos esses problemas estão em co-NP.

P, NP e co-NP



P $\neq$ NP?

NP $\cap$ co-NP $\neq$ P?

NP $\neq$ co-NP?

Redução polinomial

Permite comparar o "grau de complexidade" de problemas diferentes.

Uma redução de um problema Π a um problema Π' é um algoritmo ALG que resolve Π usando uma subrotina hipotética ALG$'$ que resolve Π', de tal forma que, se ALG$'$ é um algoritmo polinomial, então ALG é um algoritmo polinomial.

$\Pi \leq_P \Pi' =$ existe uma redução de Π a Π'.

Se $\Pi \leq_P \Pi'$ e Π' está em P, então Π está em P.

Exemplo

$\Pi =$ encontrar um ciclo hamiltoniano
$\Pi' =$ existe um ciclo hamiltoniano?

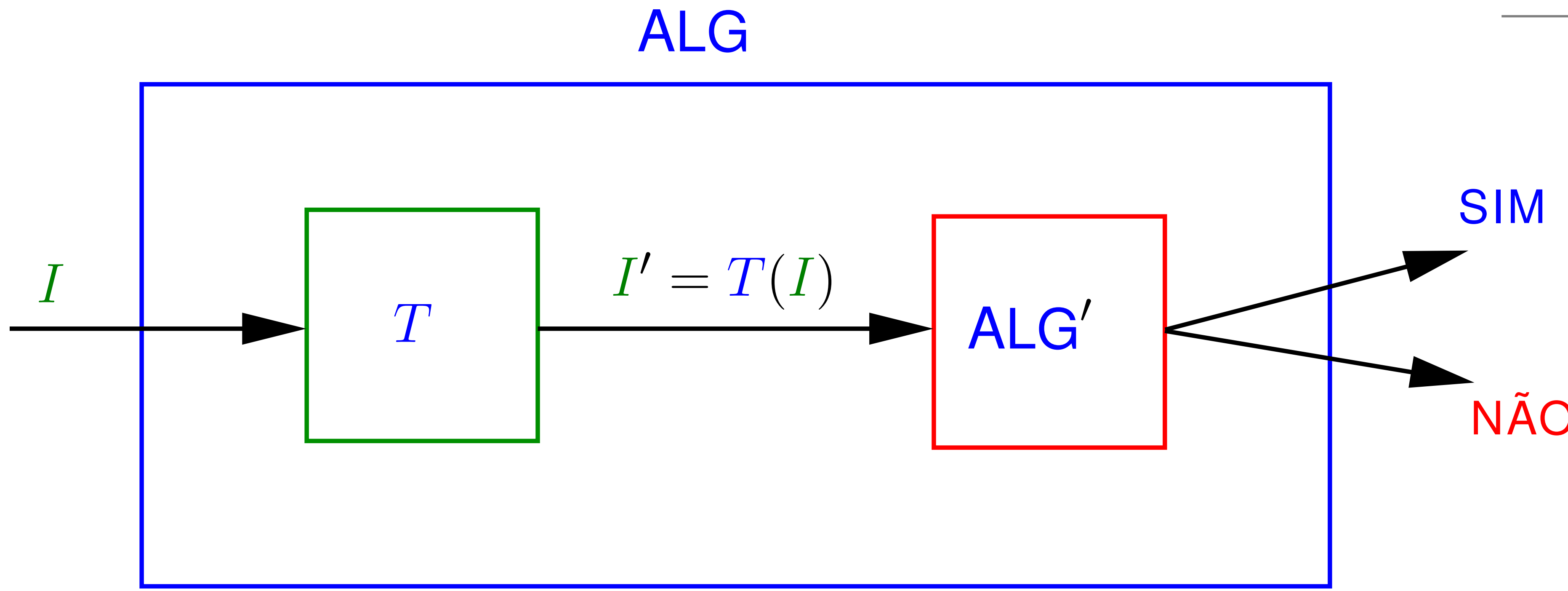
Redução de Π a Π': ALG$'$ é um algoritmo que resolve Π'

ALG (G)

se ALG$'(G) =$ NÃO
    então devolva "G não é hamiltoniano"
para cada aresta uv de G **faça**
    $H \leftarrow G - uv$
    se ALG$'(H) =$ SIM
        então $G \leftarrow G - uv$
devolva G

Se ALG$'$ consome tempo $\mathrm{O}(p(n))$, então ALG consome tempo $\mathrm{O}(m\, p(\langle G \rangle))$, onde $m =$ número de arestas de G.

Esquema comum de reduções



Faz apenas uma chamada ao algoritmo ALG$'$.

T transforma uma instância I de Π em uma instância $I' = T(I)$ de Π' tal que

$$\Pi(I) = \text{SIM se e somente se } \Pi'(I') = \text{SIM}$$

T é uma espécie de “filtro” ou “compilador”.

Satisfatibilidade

Problema: Dada um fórmula booleana ϕ nas variáveis $x_1, \ldots, x_n$, existe uma atribuição

$$t : \{x_1, \ldots, x_n\} \rightarrow \{\text{VERDADE}, \text{FALSO}\}$$

que torna ϕ verdadeira?

Exemplo:

$$\phi = (x_1) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_3)$$

Se $t(x_1) = \text{VERDADE}, t(x_2) = \text{FALSO}, t(x_3) = \text{FALSO}$, então $t(\phi) = \text{VERDADE}$

Se $t(x_1) = \text{VERDADE}, t(x_2) = \text{VERDADE}, t(x_3) = \text{FALSO}$, então $t(\phi) = \text{FALSO}$

Sistemas lineares 0-1

Problema: Dadas uma matriz A e um vetor b,

$$Ax \geq b$$

possui uma solução tal que $x_i = 0$ ou $x_i = 1$ para todo i?

Exemplo:

$$\begin{array}{rrrrrrrr} & x_1 & & & & & \geq & 1 \\ - & x_1 & - & x_2 & + & x_3 & \geq & -1 \\ & & & & - & x_3 & \geq & 0 \end{array}$$

tem uma solução 0-1?

Sim! $x_1 = 1, x_2 = 0$ e $x_3 = 0$ é solução.

Exemplo 1

Satisfatibilidade $\leq_P$ Sistemas lineares 0-1

A transformação T recebe uma fórmula booleana ϕ
e devolve um sistema linear $Ax \geq b$
tal que ϕ é satisfatível se e somente se o sistema $Ax \geq b$ admite uma solução 0-1.

Exemplo:

$$\phi = (x_1) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_3)$$

$$\begin{array}{ccccccc} x_1 & & & & & \geq & 1 \\ 1 - x_1 & + & 1 - x_2 & + & x_3 & \geq & 1 \\ & & & & 1 - x_3 & \geq & 1 \end{array}$$

Exemplo 2

Verifique que

Ciclo hamiltoniano $\leq_P$ Caminho hamiltoniano entre u e v

Verifique que

Caminho hamiltoniano entre u e v $\leq_P$ Caminho hamiltoniano