Problemas NP-Completos

Fábio Botler

Programa de Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro

Aula passada

- Reduções de Karp $\times$ Reduções de Cook
- Restrições e Extensões de Problemas

Aula de hoje

- Algoritmos superpolinomiais
- Isomorfismo de Grafos
- Algoritmos pseudopolinomiais
- Problemas numéricos

Definição

Um algoritmo é **superpolinomial** *se sua complexidade é* $\omega(n^c)$ *para toda constante* c.

Definição

Um algoritmo é **exponencial** *se sua complexidade é* $2^{O(n^c)}$ *para alguma constante* c.

Definição (sem consenso)

Um algoritmo é **subexponencial** *se sua complexidade é* $2^{o(n)}$.

Definição

Um algoritmo é **quasipolinomial** *quando sua complexidade é* $2^{O((\log n)^c)}$ *para alguma constante* c.

Isomorfismo de Grafos

DADOS:	Um grafo G e um grafo H
OBJETIVO:	G é isomorfo a H?

Isomorfismo de Grafos

DADOS: Um grafo G e um grafo H
OBJETIVO: G é isomorfo a H?

▷ `Isomorfismo de Grafos` é **restrição** de `Isomorfismo de subgrafos` (que é NP-Completo)

`Isomorfismo de Grafos`

DADOS: Um grafo G e um grafo H
OBJETIVO: G é isomorfo a H?

$\triangleright$ `Isomorfismo de Grafos` é **restrição** de `Isomorfismo de subgrafos` (que é NP-Completo)

Teorema (Babai, 2015+)

Isomorfismo de Grafos pode ser resolvido em tempo quasipolinomial.

Algoritmos Pseudopolinomiais

Definição

Um algoritmo é **pseudopolinomial** *se sua complexidade for polinomial no tamanho da instância quando codificada em unário.*

Definição

Um algoritmo é **pseudopolinomial** *se sua complexidade for polinomial no tamanho da instância quando codificada em unário.*

Valores $\times$ Tamanho

Definição

Um algoritmo é **pseudopolinomial** *se sua complexidade for polinomial no tamanho da instância quando codificada em unário.*

$$\text{Valores} \times \text{Tamanho}$$

Exemplo

Algoritmo natural para `Fatorização de Inteiros`*:* $O(n)$

Definição

Um algoritmo é **pseudopolinomial** *se sua complexidade for polinomial no tamanho da instância quando codificada em unário.*

Valores $\times$ Tamanho

Exemplo

Algoritmo natural para `Fatorização de Inteiros`: $O(n)$

Exemplo

Algoritmo de programação dinâmica para o `Problema da Mochila`: $O(nW)$

Definição

Um problema NP-Completo que admite algoritmo pseudopolinomial é chamado **NP-Completo fraco***.*

Definição

Um problema NP-Completo que admite algoritmo pseudopolinomial é chamado **NP-Completo fraco**.

Exemplo

O `Problema da Mochila`

Exemplo

`Circuito Hamiltoniano:`

- ▷ *o valor do maior dado de entrada (índices de vértices/arestas) é polinomial no tamanho do grafo.*
- ▷ *Representar tais dados em forma unária não muda a natureza do tamanho dos dados, i.e., os dados continuam polinomiais.*

Exemplo

Circuito Hamiltoniano:

- ▷ *o valor do maior dado de entrada (índices de vértices/arestas) é polinomial no tamanho do grafo.*
- ▷ *Representar tais dados em forma unária não muda a natureza do tamanho dos dados, i.e., os dados continuam polinomiais.*
- ▷ *Um algoritmo pseudopolinomial, nesse caso, é um algoritmo polinomial.*

- Há problemas NP-Completos para os quais a existência de algoritmos pseudopolinomiais implica em $P = NP$.

- ▷ Há problemas NP-Completos para os quais a existência de algoritmos pseudopolinomiais implica em $P = NP$.

Definição

Um problema NP-Completo para o qual a existência de um algoritmo pseudopolinomial que o decida implica na existência de um algoritmo polinomial que o decida é chamado **NP-Completo forte**.

- ▷ Há problemas NP-Completos para os quais a existência de algoritmos pseudopolinomiais implica em $P = NP$.

Definição

Um problema NP-Completo para o qual a existência de um algoritmo pseudopolinomial que o decida implica na existência de um algoritmo polinomial que o decida é chamado **NP-Completo forte***.*

- ▷ Todos os problemas com a "natureza" do Problema `Circuito Hamiltoniano` são NP-Completos fortes

Caixeiro-viajante

DADOS: Um grafo completo G com pesos $w\colon E(G) \to \mathbb{N}$, e um inteiro k

OBJETIVO: G possui um circuito Hamiltoniano com peso no máximo k?

Caixeiro-viajante

DADOS: Um grafo completo G com pesos $w\colon E(G) \to \mathbb{N}$, e um inteiro k

OBJETIVO: G possui um circuito Hamiltoniano com peso no máximo k?

- ▷ É NP-Completo:
  `Circuito Hamiltoniano` $\propto$ `Caixeiro-viajante`:
  Dado grafo G, seja G' um grafo completo tal que $V(G') = V(G)$, e faça

$$w(e) = 1 \text{ se } e \in E(G)$$
$$w(e) = 2 \text{ se } e \notin E(G)$$

Caixeiro-viajante

DADOS: Um grafo completo G com pesos $w \colon E(G) \to \mathbb{N}$, e um inteiro k

OBJETIVO: G possui um circuito Hamiltoniano com peso no máximo k?

- ▷ É NP-Completo:
  `Circuito Hamiltoniano` $\propto$ `Caixeiro-viajante`:
  Dado grafo G, seja G' um grafo completo tal que $V(G') = V(G)$, e faça

$$w(e) = 1 \text{ se } e \in E(G)$$
$$w(e) = 2 \text{ se } e \notin E(G)$$

- ▷ Claramente, G possui circuito Hamiltoniano se e somente se G' possui um circuito Hamiltoniano de peso no máximo n.

Exemplo

Seja $W = \sum_{e \in E(G)} w(e)$*, e suponha que exista um algoritmo* A *de complexidade* $O\big((nWk)^c\big)$ *que decida* `Caixeiro-viajante`*, para algum* $c \in \mathbb{N}$*.*

Exemplo

Seja $W = \sum_{e \in E(G)} w(e)$*, e suponha que exista um algoritmo* A *de complexidade* $O\big((nWk)^c\big)$ *que decida* `Caixeiro-viajante`*, para algum* $c \in \mathbb{N}$*.*

Esse algoritmo aplicado na instância obtida (G', w, n) *é polinomial no tamanho de* G*, pois* W *e* n *são polinomiais em* n*.*

Exemplo

Seja $W = \sum_{e \in E(G)} w(e)$*, e suponha que exista um algoritmo* A *de complexidade* $O\big((nWk)^c\big)$ *que decida* `Caixeiro-viajante`*, para algum* $c \in \mathbb{N}$*.*

Esse algoritmo aplicado na instância obtida (G', w, n) *é polinomial no tamanho de* G*, pois* W *e* n *são polinomiais em* n*.*

Logo, isso implica $P = NP$*.*

Exemplo

Seja $W = \sum_{e \in E(G)} w(e)$*, e suponha que exista um algoritmo* A *de complexidade* $O\big((nWk)^c\big)$ *que decida* `Caixeiro-viajante`*, para algum* $c \in \mathbb{N}$*.*

Esse algoritmo aplicado na instância obtida (G', w, n) *é polinomial no tamanho de* G*, pois* W *e* n *são polinomiais em* n*.*

Logo, isso implica $P = NP$*.*

Portanto, `Caixeiro-viajante` *é NP-Completo forte.*

Um algoritmo pseudopolinoimal é polinomial para instâncias nas quais o valor do maior dado é polionomial no tamanho da instância.

Um algoritmo pseudopolinoimal é polinomial para instâncias nas quais o valor do maior dado é polionomial no tamanho da instância.

Exemplo

O Problema da Mochila:

Um algoritmo pseudopolinoimal é polinomial para instâncias nas quais o valor do maior dado é polionomial no tamanho da instância.

Exemplo

O `Problema da Mochila`*:*

Se o peso total W for polinomial no tamanho da instância, i.e., $W = O(n^c)$, então o algoritmo $O(nW)$ é $O(n^{c+1})$.

Problemas Numéricos

Definição

Um problema no qual o valor do maior dado de uma instância não é necessariamente polinomial no tamanho da instância é chamado de **Problema Numérico***.*

Definição

Um problema no qual o valor do maior dado de uma instância não é necessariamente polinomial no tamanho da instância é chamado de **Problema Numérico**.

Exemplo

`Problema da Mochila`

Definição

Um problema no qual o valor do maior dado de uma instância não é necessariamente polinomial no tamanho da instância é chamado de **Problema Numérico**.

Exemplo

Problema da Mochila

Exemplo

Fatorização de Inteiros

Como mostramos que um problema Π é NP-Completo fraco?

Como mostramos que um problema Π é NP-Completo fraco?

- $\triangleright$ Basta mostrar um algoritmo pseudopolinomial que o resolva

Como mostramos que um problema Π é NP-Completo fraco?

- $\triangleright$ Basta mostrar um algoritmo pseudopolinomial que o resolva

Como mostramos que um problema Π é NP-Completo forte?

Como mostramos que um problema Π é NP-Completo fraco?

- $\triangleright$ Basta mostrar um algoritmo pseudopolinomial que o resolva

Como mostramos que um problema Π é NP-Completo forte?

- $\triangleright$ Mostramos que se houver algoritmo pseudopolinomial para Π então existe algoritmo polinomial para um problema NP-Completo

Como mostramos que um problema Π é NP-Completo fraco?

- $\triangleright$ Basta mostrar um algoritmo pseudopolinomial que o resolva

Como mostramos que um problema Π é NP-Completo forte?

- $\triangleright$ Mostramos que se houver algoritmo pseudopolinomial para Π então existe algoritmo polinomial para um problema NP-Completo

- $\triangleright$ Equivalentemente, mostramos que há uma restrição de Π que é NP-Completa

Subconjunto Soma

DADOS:	Um conjunto finito $S \subseteq \mathbb{N}$ e um inteiro t
OBJETIVO:	existe $S' \subseteq S$ tal que $\sum_{s \in S'} s = t$?

Subconjunto Soma

DADOS:	Um conjunto finito $S \subseteq \mathbb{N}$ e um inteiro t
OBJETIVO:	existe $S' \subseteq S$ tal que $\sum_{s \in S'} s = t$?

- ▷ É um problema numérico, pois o valor de $s \in S$ não é limitado pelo tamanho de S.

Teorema

Subconjunto Soma é NP-Completo.

Teorema

Subconjunto Soma é NP-Completo.

Prova

- ▷ *Cobertura por Vértices* $\propto$ *Subconjunto Soma*

Teorema

Subconjunto Soma é NP-Completo.

Prova

- $\triangleright$ *Cobertura por Vértices* $\propto$ *Subconjunto Soma*
- $\triangleright$ *Seja* (G, k) *uma instância do problema Cobertura por Vértices.*
- $\triangleright$ *Ordene as arestas e fixe os índices delas*

Teorema

Subconjunto Soma é NP-Completo.

Prova

- *Cobertura por Vértices* $\propto$ *Subconjunto Soma*
- *Seja* (G, k) *uma instância do problema Cobertura por Vértices.*
- *Ordene as arestas e fixe os índices delas*
- *Para a i-ésima aresta* e_i, *seja* $s_{e_i} = 4^i$

Teorema

Subconjunto Soma é NP-Completo.

Prova

- ▷ *Cobertura por Vértices* $\propto$ *Subconjunto Soma*
- ▷ *Seja (G, k) uma instância do problema Cobertura por Vértices.*
- ▷ *Ordene as arestas e fixe os índices delas*
- ▷ *Para a i-ésima aresta e_i, seja $s_{e_i} = 4^i$*
- ▷ *Para $v \in V(G)$, seja $E(v)$ o conjunto de arestas incidentes a v e seja*

$$s_v = 4^m + \sum_{e \in E(v)} s_e$$

Teorema

Subconjunto Soma é NP-Completo.

Prova

- *Cobertura por Vértices* $\propto$ *Subconjunto Soma*
- *Seja (G, k) uma instância do problema Cobertura por Vértices.*
- *Ordene as arestas e fixe os índices delas*
- *Para a i-ésima aresta e_i, seja $s_{e_i} = 4^i$*
- *Para $v \in V(G)$, seja $E(v)$ o conjunto de arestas incidentes a v e seja*

$$s_v = 4^m + \sum_{e \in E(v)} s_e$$

- *Seja $t = k4^m + \sum_{e \in E(G)} 2s_e$*

Prova

- $\triangleright$ $S = \{s_x \colon x \in V(G) \cup E(G)\}$.

Prova

- $\triangleright$ $S = \{s_x \colon x \in V(G) \cup E(G)\}$.

- $\triangleright$ *G possui uma cobertura por vértices de tamanho k*
  se e somente se
  existe $S' \subseteq S$ *tal que* $\sum_{s \in S'} s = t$

Teorema

Subconjunto Soma é NP-Completo fraco.

Teorema

- ▷ *Existe algoritmo de programação dinâmica de complexidade $O\big(n(t+1)\big)$ que resolve Subconjunto Soma.*

- ▷ *Para cada $k \in \{1, \ldots, n\}$, seja $S_k = \{s_1, \ldots, s_k\}$.*

- ▷ *Temos que Subconjunto Soma(S_k, t) = SIM se e somente se Subconjunto Soma(S_{k-1}, t) = SIM ou Subconjunto Soma$(S_{k-1}, t - s_k)$ = SIM.*

Denotaremos por $x_{i,j}$ o resultado de `Subconjunto Soma`(S_i, j)

`Subconjunto Soma`(S, t)

Para cada $i = 1, \ldots, n$

  $x_{i,0} = \text{SIM}$

Para cada $j = 1, \ldots, t$

  Para cada $i = 1, \ldots, n$

    se $x_{i-1,j} = SIM$ ou $x_{i-1,j-s_j} = SIM$,

      então $x_{i,j} = SIM$

      senão $x_{i,j} = \text{NÃO}$

Retorna $x_{n,t}$

Problemas NP-Completos

Fábio Botler

Programa de Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro