Roteiro da Aula 21

1. A classe **P**
2. Exemplos de Problemas
   - P_1 e P_2
   - *PATH*
   - *Hamiltonian Path*
   - *Traveling Salesman Problem*

A classe **P**

Classe **P**

Um problema A pertence à classe **P** se A possui uma cota superior $O(n^k)$, para alguma constante k. Ou seja, se existe um algoritmo polinomial para A.

Classe **P**

Uma linguagem $\mathcal{L}$ pertence à classe **P** se $\mathcal{L}$ pode ser decidida por uma máquina de Turing de complexidade de tempo de pior caso $O(n^k)$, para alguma constante k.

Relevância da classe **P**

- Considere a diferença entre n^3 e 2^n:
  - $100^3 = 1000000$: um milhão;
  - $2^{100} = 1267650600228229401496703205376$: um virgula dois zilhões ...
- Um algoritmo polinomial pode não ser útil na prática; mas um algoritmo exponencial certamente não é útil na prática!

Relevância da classe **P**

- **P** é invariante, robusta, em relação aos modelos de computação relevantes: todos são polinomialmente equivalentes!

C	s operações		$f(n) = n^4$
ASSEMBLER	s^2 operações	x operações	$f'(n) = (n^2)^4$
Máq. Turing		x^3 operações	$f''(n) = ((n^3)^2)^4$

Mas $f''(n) = ((n^3)^2)^4 = n^{24}$, que é polinomial!

Exemplos de Problemas

Problema Linguagem	Algoritmo MT	função de custo de pior caso	cota superior notação assintótica
$\mathcal{L}_1$	M_1 M_2	$\frac{n^2}{2} + n + 2$ $2n \log n$	$O(n^2)$ $\Theta(n \log n)$, $O(n^2)$
P_1			
P_2			
PATH			
HP			
TSP			

P_1

- Dado um conjunto C de n números inteiros, positivos ou negativos, e um inteiro positivo s, decidir se existe um subconjunto de C cuja soma de seus elementos é maior ou igual a s.

- Exemplo: $C = \{-1, -7, 3, 11, -2, 0, 4, -8\}$; $s = 16$

P_1

- Dado um conjunto C de n números inteiros, positivos ou negativos, e um inteiro positivo s, decidir se existe um subconjunto de C cuja soma de seus elementos é maior ou igual a s.

- Exemplo: $C = \{-1, -7, 3, 11, -2, 0, 4, -8\}$; $s = 16$
- Resposta: sim.

Quais algoritmos temos para esse problema?

P_2

- Dado um conjunto C de n números inteiros, positivos ou negativos, decidir se existe um subconjunto de C cuja soma de seus elementos é zero.

- Exemplo: $C = \{-1, -7, 3, 11, -2, 5, 4, -8\}$;

P_2

- Dado um conjunto C de n números inteiros, positivos ou negativos, decidir se existe um subconjunto de C cuja soma de seus elementos é zero.

- Exemplo: $C = \{-1, -7, 3, 11, -2, 5, 4, -8\}$;
- Resposta: sim.

Quais algoritmos temos para esse problema?

Conclusão

- $P_1 \in$ **P**;
- Não sabemos se $P_2 \in$ **P**. É decidível, recursivo, mas ainda não conhecemos nenhum algoritmo polinomial para ele...

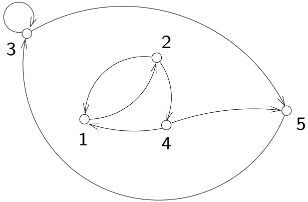
Exemplos de Problemas

Problema Linguagem	Algoritmo MT	função de custo de pior caso	cota superior notação assintótica
$\mathcal{L}_1$	M_1 M_2	$\frac{n^2}{2}+n+2$ $2n\log n$	$O(n^2)$ $\Theta(n\log n)$, $O(n^2)$
P_1	Força Bruta Linear		$\Theta(n2^n)$ $\Theta(n)$
P_2	Força Bruta		$\Theta(n2^n)$
PATH			
HP			
TSP			

Grafos

- Um dígrafo (grafo orientado) é uma tupla $G = (V, E)$ onde:
  - V é um conjunto finito de vértices;
  - $E \subseteq V \times V$ é o conjunto de arestas;
- Exemplo: $G = (\{1, 2, 3, 4, 5\}, \{(1, 2), (2, 1), (2, 4), (4, 1), (4, 5), (3, 5), (5, 3), (3, 3)\})$.

Representação Gráfica



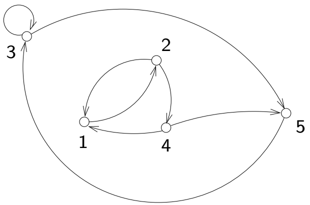
PATH

- Dado um dígrafo $G = (V, E)$ e um par de vértices (s, t), decidir se existe um caminho em G de s até t.
- A entrada do problema é uma Matriz de Adjacência:

	1	2	3	4	5
1	0	1	0	0	0
2	1	0	0	1	0
3	0	0	1	0	1
4	1	0	0	0	1
5	0	0	1	0	0

A matriz tem n^2 posições,
que é o número máximo de arestas num dígrafo
com n vértices

Algoritmo Força Bruta



- Para cada permutação de V:
  - Verificar, usando a matriz, se existe caminho $s \rightarrow t$ na permutação.

Exemplo: $s = 1$ e $t = 5$

Teoria da Computação
116360
Aula 21

Roteiro

A classe **P**

Exemplos de Problemas

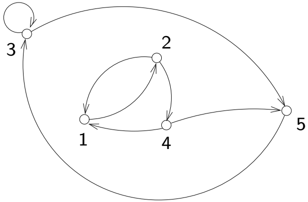
P_1 e P_2

PATH

Hamiltonian Path

Traveling Salesman Problem

Algoritmo Força Bruta

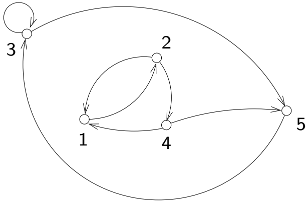


- Para cada permutação de V:
  - Verificar, usando a matriz, se existe caminho $s \rightarrow t$ na permutação.

Exemplo: $s = 1$ e $t = 5$

Custo $\Theta(nn!)$, onde n é o número de vértices

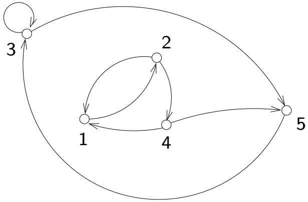
Busca em Largura



- Marcar vértice s;
- Repetir até que nenhum novo vértice seja marcado:
  - Para cada aresta (a, b) de G:
    - Se a está marcado e b não está marcado, marcar b.
- Verificar se t está marcado;

Exemplo: $s = 1$ e $t = 5$

Busca em Largura



- Marcar vértice s;
- Repetir até que nenhum novo vértice seja marcado:
  - Para cada aresta (a, b) de G:
    - Se a está marcado e b não está marcado, marcar b.
- Verificar se t está marcado;

Exemplo: $s = 1$ e $t = 5$

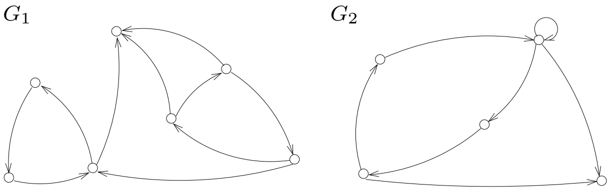
Custo $O(n^3)$

Exemplos de Problemas

Problema Linguagem	Algoritmo MT	função de custo de pior caso	cota superior notação assintótica
$\mathcal{L}_1$	M_1 M_2	$\frac{n^2}{2} + n + 2$ $2n \log n$	$O(n^2)$ $\Theta(n \log n)$, $O(n^2)$
P_1	Força Bruta Linear		$\Theta(n2^n)$ $\Theta(n)$
P_2	Força Bruta		$\Theta(n2^n)$
PATH	Força Bruta B. em Largura		$\Theta(nn!)$ $\Theta(n^3)$
HP			
TSP			

Hamiltonian Path

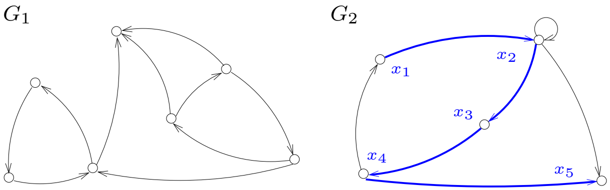
- Dado um dígrafo $G = (V, E)$, decidir se existe um caminho passando por todos os vértices, sem repetir vértice. Um caminho hamiltoniano, então, é uma permutação $(x_1, x_2, x_3, \dots, x_n)$ do conjunto de vértices, tal que existe aresta (x_i, x_{i+1}), $1 \leq i \leq n - 1$.



A força bruta continua funcionando ...

Hamiltonian Path

- Dado um dígrafo $G = (V, E)$, decidir se existe um caminho passando por todos os vértices, sem repetir vértice. Um caminho hamiltoniano, então, é uma permutação $(x_1, x_2, x_3, \ldots, x_n)$ do conjunto de vértices, tal que existe aresta (x_i, x_{i+1}), $1 \leq i \leq n-1$.



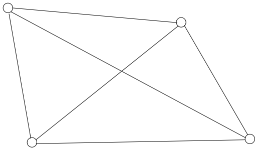
A força bruta continua funcionando ...

Exemplos de Problemas

Problema Linguagem	Algoritmo MT	função de custo de pior caso	cota superior notação assintótica
$\mathcal{L}_1$	M_1	$\frac{n^2}{2} + n + 2$	$O(n^2)$
	M_2	$2n \log n$	$\Theta(n \log n)$, $O(n^2)$
P_1	Força Bruta		$\Theta(n2^n)$
	Linear		$\Theta(n)$
P_2	Força Bruta		$\Theta(n2^n)$
PATH	Força Bruta		$\Theta(nn!)$
	B. em Largura		$\Theta(n^3)$
HP	Força Bruta		$\Theta(nn!)$
TSP			

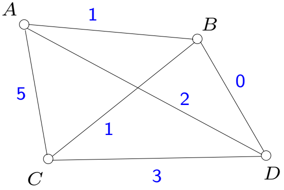
Grafo não-direcionado

- Um grafo não-direcionado formalmente é um dígrafo $G = (V, E)$ onde:
  - Se $(a, b) \in E$, então:
    - $a \neq b$;
    - $(b, a) \in E$.
- Exemplo:



Grafo não-direcionado

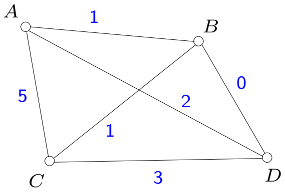
- Dado um grafo não-direcionado $G = (V, E)$, um circuito é uma permutação $(x_1, x_2, x_3, \ldots, x_n)$ do conjunto de vértices, tal que existe aresta (x_i, x_{i+1}), $1 \leq i \leq n-1$, e aresta (x_n, x_1).
- Uma função que associa custos às arestas de um grafo pode ser representada juntamente com a matriz de adjacência:



	A	B	C	D
A	-	1	5	2
B	1	-	1	0
C	5	1	-	3
D	2	0	3	-

Traveling Salesman Problem

- Dado um grafo não-direcionado, com custos inteiros não-negativos nas arestas, e um inteiro s, decidir se existe um circuito de custo total menor ou igual a s.



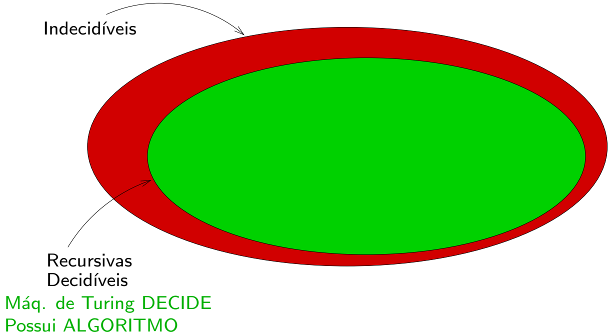
De novo, a força bruta continua funcionando ...

Exemplos de Problemas

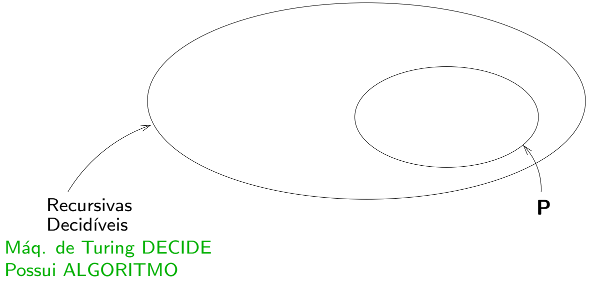
Problema Linguagem	Algoritmo MT	função de custo de pior caso	cota superior notação assintótica
$\mathcal{L}_1$	M_1	$\frac{n^2}{2} + n + 2$	$O(n^2)$
	M_2	$2n \log n$	$\Theta(n \log n)$, $O(n^2)$
P_1	Força Bruta		$\Theta(n2^n)$
	Linear		$\Theta(n)$
P_2	Força Bruta		$\Theta(n2^n)$
PATH	Força Bruta		$\Theta(nn!)$
	B. em Largura		$\Theta(n^3)$
HP	Força Bruta		$\Theta(nn!)$
TSP	Força Bruta		$\Theta(nn!)$

Têm ou não têm algoritmo polinomial?

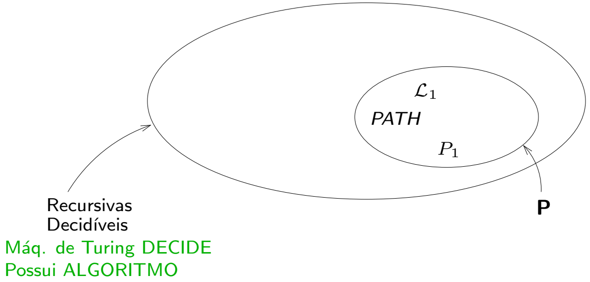
Situação Atual



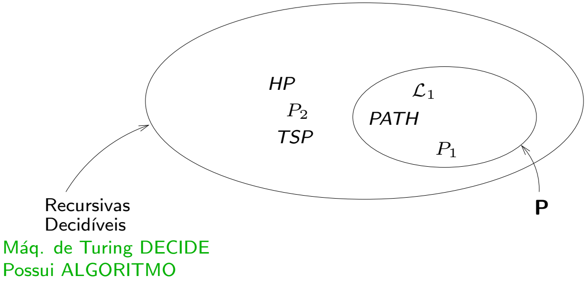
Situação Atual



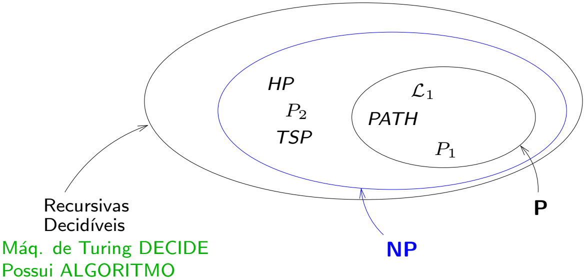
Situação Atual



Situação Atual



Situação Atual



Que classe **NP** é essa?

Por que os americanos estão pagando US$ 1,000,000 para quem provar que $\mathbf{NP} = \mathbf{P}$ ou $\mathbf{NP} \neq \mathbf{P}$?