

AULA 23

Complexidade computacional:

P versus NP

CLR 36 ou CLRS 34

Consumo de tempo de um *algoritmo*

Complexidade de um *problema*

Problemas e instâncias

Instâncias = cadeias de caracteres

N = tamanho de uma instância

Exemplos

- problema da subsequência crescente máxima
- problema subcoleção disjunta máxima de intervalos
- problema da mochila
- problema da árvore geradora de peso mínimo

Complexidade de um problema

Problema **polinomial**:
existe algoritmo $O(N^i)$ para algum i

Por que polinômios?

Classe **P** de problemas

Para muitos problemas, não se conhece algoritmo melhor que “testar todas as possibilidades”.
(Exemplo: examinar todas as 2^n seqües de n *bits*.)
Em geral, isso não está em P.

Problemas **de decisão**: respostas SIM ou NÃO

- subsequência crescente $\geq k$
- subcoleção disjunta $\geq k$ de intervalos
- mochila de valor $\geq k$
- árvore geradora de peso $\leq k$

Verificação de soluções

Certificados de respostas SIM

Certificados de respostas NÃO

- existe subsequência crescente $\geq k$?
- existe subcoleção disjunta $\geq k$ de intervalos?
- existe mochila de valor $\geq k$?
- grafo tem árvore geradora de peso $\leq k$?
- grafo tem circuito de comprimento $\geq k$?
- grafo tem conjunto independente de tamanho $\geq k$?
- grafo tem emparelhamento de tamanho $\geq k$?

Certificados polinomiais

Classe **NP**:

resposta SIM tem certificado polinomial

Não confunda **NP** com “não-polinomial”

Classe **coNP**:

resposta NÃO tem certificado polinomial

Fácil: $\mathbf{P} \subseteq \mathbf{NP}$ $\mathbf{P} \subseteq \mathbf{coNP}$

Ninguém sabe: $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$

Redução entre problemas

Converta instância de problema A em instância de B .
Resolva instância de B .
Converta solução de B em solução de A .

Redução polinomial

Permite comparar o “grau de complexidade”
de problemas diferentes.

Problemas **completos** em **NP**

A é completo em **NP** se todo problema em **NP** pode ser polinomialmente reduzido a A

Classe **NPC**

Classe dos problemas “mais difíceis” de **NP**

$$\mathbf{P} = \mathbf{NP} \quad \Leftrightarrow \quad \mathbf{P} \cap \mathbf{NPC} \neq \emptyset$$

Teorema de S. Cook: $\mathbf{NPC} \neq \emptyset$

- mochila
- circuito hamiltoniano
- caminho máximo
- escalonamento de tarefas
- clique máximo

TAREFA 22

Exercício 22.A

Suponha que os algoritmos A e B transformam cadeias de caracteres em outras cadeias de caracteres. O algoritmo A consome $O(n^2)$ unidades de tempo e o algoritmo B consome $O(n^4)$ unidades de tempo, onde n é o número de caracteres da cadeia de entrada. Considere agora o algoritmo AB que consiste na composição de A e B , com B recebendo como entrada a saída de A . Qual o consumo de tempo de AB ?