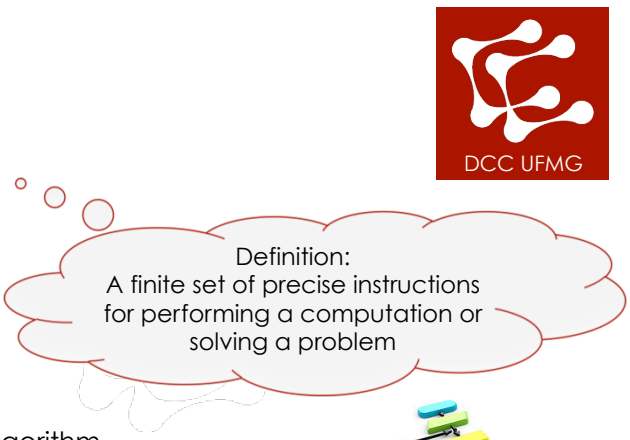
ALG I 2020-01

Algoritmos I

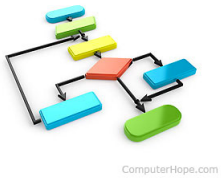
Prof. Dr. Olga Goussevskaia
Computer Science Department
Universidade Federal de Minas Gerais

Algorithm



Definition:
A finite set of precise instructions for performing a computation or solving a problem

- Synonyms for an algorithm
  - Program
  - Procedure
  - Recipe
  - …



Complexity of algorithms

DCC UFMG

- A given problem can be solved by many different algorithms...

- Which algorithms will be useful in practice?
  - They have to be correct: respect a given specification, no inf. Loop, etc
  - They have to be efficient...

Complexity of algorithms: How to define efficiency?

DCC UFMG

- Proposed Definition 1: *An algorithm is efficient if, when implemeted, it runs quickly on real instances.*
  - Where and how well we implement an algorithm?
  - How well does it scale with input size?
  - What is a real instance? Average case? Special case? Random?
- Proposed Definition 2: *An algorithm is efficient if it achieves qualitatively better worst-case performance at analytical level, than brute-force search.*
  - Nearly always contain a valuable heuristic idea that reveals parts of the instrinsic structure of a problem
  - But what is "qualitatively better performance"?
- Proposed Definition 3: *An algorithm is:*
  - *Efficient: has polynomial time for ALL inputs*
  - *Ineficcient: has exponential time for SOME inputs*
  - *Independent of platform, instance and predicts scaling properties*

Is exponential time really that bad?

Table 1: Comparing polynomial and exponential time complexity. Assume a problem of size one takes 0.000001 seconds (1 microsecond).

	Size *n*					
	10	**20**	**30**	**40**	**50**	**60**
n	0.00001 second	0.00002 second	0.00003 second	0.00004 second	0.00005 second	0.00006 second
n^2	0.0001 second	0.0004 second	0.0009 second	0.0016 second	0.0025 second	0.0036 second
n^3	0.001 second	0.008 second	0.027 second	0.064 second	0.125 second	0.216 second
n^5	0.1 second	3.2 second	24.3 second	1.7 minutes	5.2 minutes	13.2 minutes
2^n	0.001 second	1.0 second	17.9 minutes	12.7 days	35.7 years	366 centuries
3^n	0.059 second	58 minutes	6.5 years	3855 centuries	$2*10^8$ centuries	$1.3*10^{13}$ centuries

(from M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman, New York, 1979.)

What if we buy more state-of-the art hardware?

Table 2: Effect of improved technology on several polynomial and exponential time algorithms. The following table represents the size of the largest problem instance solvable in 1 hour.

Time Complexity function	**With present computer**	**With computer 100 times faster**	**With computer 1000 times faster**
n	N1	100 N1	1000 N1
n^2	N2	10 N2	31.6 N2
n^3	N3	4.46 N3	10 N3
n^5	N4	2.5 N4	3.98 N4
2^n	N5	N5+6.64	N5+9.97
3^n	N6	N6+4.19	N6+6.29

(from M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman, New York, 1979.)

Asymptotic complexity

- Let ***T(n)*** be a measure of the time required to execute an algorithm on a problem of size ***n***
- *If* ***T(n) = 2n⁴ + 3n³ + 4n² + 5n + 6****, then*

$$T(n) = n^4\left(2 + \frac{3}{n} + \frac{4}{n^2} + \frac{5}{n^3} + \frac{6}{n^4}\right)$$

And for large ***n*** *we have*

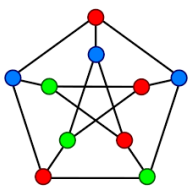
$$T(n) \cong 2n^4$$

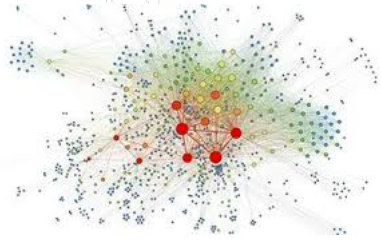
Theory people are not very interested in constants ☺

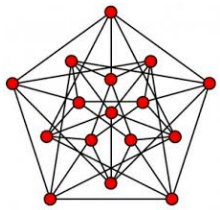
- *We say that T(n) has* ***growth of order n⁴: O(n⁴)***
- *We say that one algorithm is* ***more efficient*** *than another if its* ***worst case*** *running time has a lower* ***growth order****.*

Graph algorithms

- Problem modeling techniques
- Solution strategies



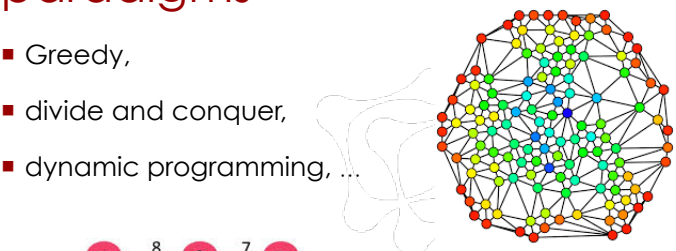


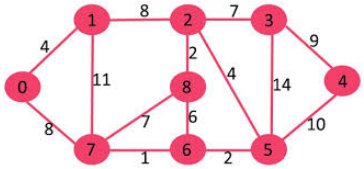


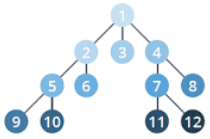
Algorithms design paradigms

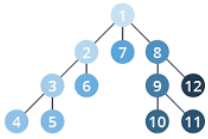
DCC UFMG

- Greedy,
- divide and conquer,
- dynamic programming, ...





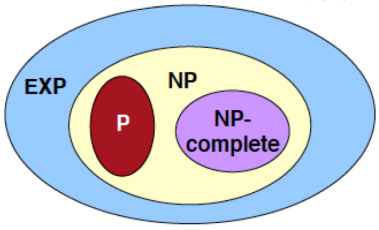




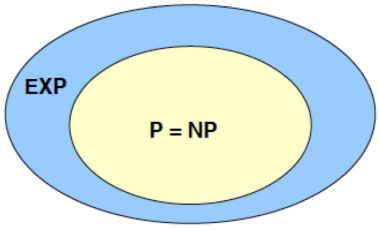
Problem complexity classes: P, NP, NP-Complete

DCC UFMG

- **NP-complete**: nobody has found an efficient algorithm for any of these problems, but, if one day someone does, then all problems in NP will be solved in polynomial time, i.e., P=NP.



If $P \neq NP$



If $P = NP$

- $1M question: Is the original (decision) problem as easy as certification?

Coping with NP-completeness

- If a problem is NP-complete (and P≠NP), an algorithm can do **at most 2** of the following **3 things**:
  1. Solve the problem exactly
  2. Guarantee to solve the problem in polynomial time
  3. Solve arbitrary instances of the problem

Estrutura do curso

Algoritmos I
2020-01

Nova grade curricular

DCC UFMG

- Antigamente:
  - AEDS I, II, III

1. Programação e Desenvolvimento de Software 1 (PDS1): conceitos básicos de programação, ex: variáveis, alocação de memória, arquivos
2. Programação e Desenvolvimento de Software 2 (PDS2): boas práticas de programação: orientação a objetos, testes/depuração, segurança
3. Estruturas de dados (ED): Métricas de complexidade, métodos de ordenação, listas, pilhas, árvores, heaps, hashs
4. Algoritmos 1 (ALG1): Algoritmos canônicos, foco em análise
5. Algoritmos 2 (ALG2): Algoritmos avançados, foco em análise

Algoritmos 1: Ementa

DCC UFMG

- Análise de complexidade (revisão)
- Prova de corretude de um algoritmo
  - Problema do casamento estável
- Algoritmos em Grafos
  - Definições e implementação
  - Busca em largura e profundidade
  - Ordenação topológica e componentes
  - Caminhos mínimos
  - Árvores geradoras mínimas
- Prova 1: **25/04** (sábado)

Algoritmos 1: Ementa

- Paradigmas Algorítmicos
  - Indução e recursão
  - Divisão e conquista
  - Algoritmos gulosos
  - Backtracking
  - Programação dinâmica
- Prova 2: **23/05**(sábado)

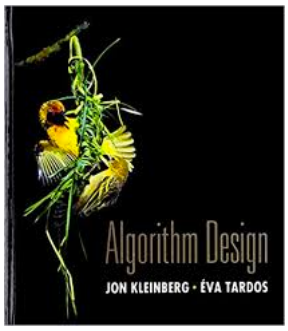
Algoritmos 1: Ementa

- NP-Completude
  - Reduções entre problemas
  - Classe P
  - Classe NP
  - Classe NP-Completo
- Prova 3: **06/06 ou 20/06** (sábado)

Algoritmos 1: Livros Texto e Avaliações

DCC UFMG

- Algorithm Design, Jon Kleinberg e Eva Tardos
  - Algoritmos: teoria e prática, T.H. Cormen et al.
  - Projeto de Algoritmos, Nívio Ziviani
- Provas aos sábados (3x 20 ptos)
  - **25/04** ??h: Algoritmos em grafos
  - **23/05** ??h: Paradigmas algorítmicos
  - **06/06 ou 20/06** ??h: NP-Completude
  - Não terá suplementar*
- Trabalhos Práticos
  - TP0 (10 ptos):
  - TP1 (15 ptos):
  - TP2 (15 ptos):



Conclusão

Algoritmos I
2020-01

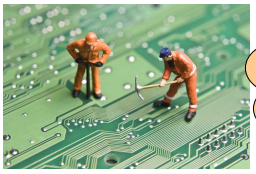
DCC UFMG

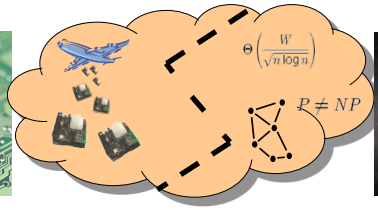
PRACTICE x THEORY: Why so little interaction?

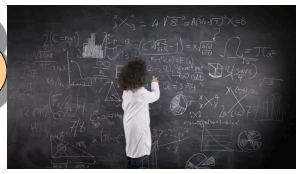
DCC UFMG

Theory is useless...

Practice is trivial...







SYSTEMS

THEORY

DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO
UFMG

ALG I 2020-01

Algoritmos I

Prof. Dr. Olga Goussevskaia
Computer Science Department
Universidade Federal de Minas Gerais