# BCC204 - Teoria dos Grafos

Marco Antonio M. Carvalho

Departamento de Computação
Instituto de Ciências Exatas e Biológicas
Universidade Federal de Ouro Preto


# Conteúdo

## Fonte

Este material é baseado no livro

- Goldbarg, M., & Goldbarg, E. (2012). *Grafos: conceitos, algoritmos e aplicações*. Elsevier.

## Licença

**Aviso!**

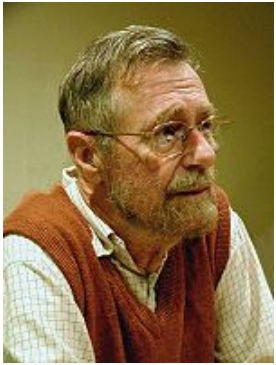
Esta apresentação é uma introdução informal à questão P vs. NP.

Não há a intenção de ser totalmente precisa historicamente e tecnicamente.

Este tópico será estudado com profundidade em diferentes disciplinas do curso de graduação.

Baseado no artigo "The Status of the P Versus NP Problem".

# Problemas Intratáveis



**Edsger Dijkstra,Prêmio Turing em 1972**

"Ciência da computação tem tanto a ver com o computador como a Astronomia com o telescópio, a Biologia com o microscópio, ou a Química com os tubos de ensaio. A Ciência não estuda ferramentas, mas o que fazemos e o que descobrimos com elas."

## Introdução

Quase todos algoritmos vistos até aqui possuem tempo polinomial: em entradas de tamanho $n$, o tempo de execução no pior caso é $O(n^k)$, para alguma constante $k$.

Podemos pensar, intuitivamente, que todos os problemas podem ser resolvidos em tempo polinomial.

Entretanto, a resposta é não. Há problemas que não podem ser resolvidos pelo computador, e outros que podem, porém, não em tempo $O(n^k)$, para alguma constante $k$.

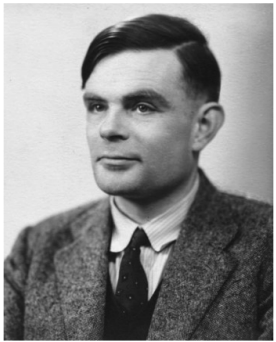
## Introdução

Geralmente, designamos os problemas resolvíveis em tempo polinomial como *tratáveis*, ou *fáceis*.

Por outro lado, designamos os problemas resolvíveis apenas em tempo supra polinomial como *intratáveis*, ou *difíceis*.

Um problema é considerado **intratável** caso não haja algoritmo em tempo polinomial que o resolva deterministicamente.

Veremos alguns exemplos mais à frente.



## Alan Turing

- Matemático, lógico e criptoanalista inglês
  - Participação importante na II Guerra Mundial.
- **Pai da Ciência da Computação**
  - Dedicou a via à teoria da computabilidade.
- Formalizou os conceitos de algoritmo e computabilidade.
- Aos 24 anos, criou a Máquina de Turing.
- Parte de sua vida foi retratada no filme "O Jogo da Imitação" de 2014.
- Hoje o Prêmio Turing equivale ao Nobel da Computação.

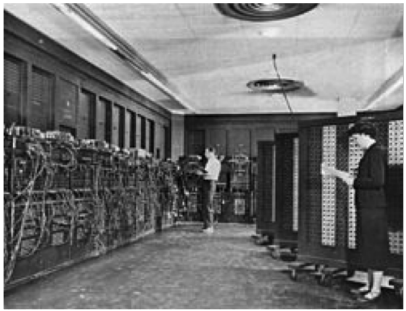
## Máquina de Turing

A **Máquina de Turing** é um modelo de computador conceitual, abstrato, também chamada de Máquina Universal.

Realiza qualquer computação matemática que possa ser representada por um algoritmo.

Capaz de computar tudo que é computável.

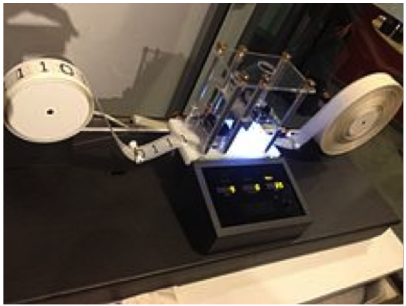
É o centro do conceito da arquitetura dos computadores modernos.



## Máquina de Turing vs. ENIAC

A primeira referência a uma máquina de Turing é de um artigo publicado em em 1936.

O ENIAC foi o primeiro computador eletrônico de propósitos variados – anunciado em 1946.



## Máquina de Turing

Existem basicamente dois tipos de Máquinas de Turing:

- Máquina de Turing **Determinística**:
  - Se 'A', então faça 'B'.
- Máquina de Turing **Não Determinística**:
  - Se 'A', então faça 'B' ou 'C' ou 'D' ou ...

## Computabilidade

Alan Turing, aos 24 anos, delineou a idéia de **computabilidade**.

Existe alguma coisa que não possa ser feita mecanicamente (ou seja, sem intuição ou inteligência)?

Em seu artigo original, Turing demonstra a existência de um **problema insolucionável** mecanicamente.

## Computabilidade

A Computabilidade e a Teoria da Complexidade Computacional estudam os **limites** da computação:

- Quais problemas jamais poderão ser resolvidos por um computador, independente da sua velocidade ou memória?
- Quais problemas podem ser resolvidos por um computador, mas requerem um período tão extenso de tempo para completar a ponto de tornar a solução impraticável?

## P

Suponhamos que temos um grande grupo de alunos e precisamos agrupá-los em **duplas compatíveis** para um trabalho:

- Nem todos os alunos são compatíveis entre si;
- Tentar todas as possibilidades não é uma alternativa;
- Se o grupo tiver 40 alunos, temos mais do que 300 bilhões de trilhões de possíveis pares.

Em 1965, Jack Edmonds criou um algoritmo eficiente para resolver este problema e ajudou a definir o que é **computação eficiente**.

## P

Computação eficiente passou a ser definida como a existência de um algoritmo que resolve um problema em tempo polinomial em relação ao tamanho da entrada.

A classe de problemas para os quais há algoritmos eficientes passou a ser conhecida como **classe P**, de "Tempo Determinístico Polinomial".

## NP

Infelizmente, para muitos problemas parece não haver algoritmo eficiente:

- E se quisermos achar o maior grupo de alunos tal que todos são compatíveis entre si (clique máximo)?
- E se quisermos sentar à mesa todos os alunos, de maneira que dois alunos incompatíveis não fiquem lado a lado (ciclo hamiltoniano)?
- E se quisermos dividir os alunos em trios, de forma que cada aluno estará sempre com outros dois incompatíveis (3 coloração)?

## NP

Todos estes problemas possuem uma propriedade em comum:

- Dada uma solução qualquer (por exemplo, o mapa das cadeiras em uma mesa), é possível conferir a solução de maneira eficiente.

É como um quebra cabeça: verificar se está montado é mais fácil do que montar.

O conjunto de problemas que possuem soluções verificáveis em tempo polinomial define a classe NP

- De "Tempo Polinomial Não Determinístico".
- Somente uma Máquina de Turing Não Determinística pode resolvê-lo em tempo determinístico polinomial.

## NP-Completo

Os mais difíceis problemas em NP formam ainda uma outra classe, a NP-Completo.

A característica notória desta classe é que um algoritmo eficiente para qualquer um dos problemas pode ser adaptado facilmente para qualquer outro problema desta classe.

Resolver **um** implica em resolver **todos**.

Vários problemas NP-completos são interessantes porque se parecem muito com problemas fáceis:

- Caminho mais curto vs. Caminho mais longo;
- Ciclo Euleriano vs. Ciclo Hamiltoniano;
- 2-SAT vs. 3-SAT.

# Problemas Intratáveis

## NP-Completo

Em 1971, Richard Karp identificou os primeiros 21 problemas da classe e contribuiu para o desenvolvimento da teoria da NP-Completude.

Posteriormente, centenas de outros problemas foram identificados por outros pesquisadores.

## NP-Completo

A maioria dos problemas de interesse pertencem comprovadamente à classe NP-Completo:

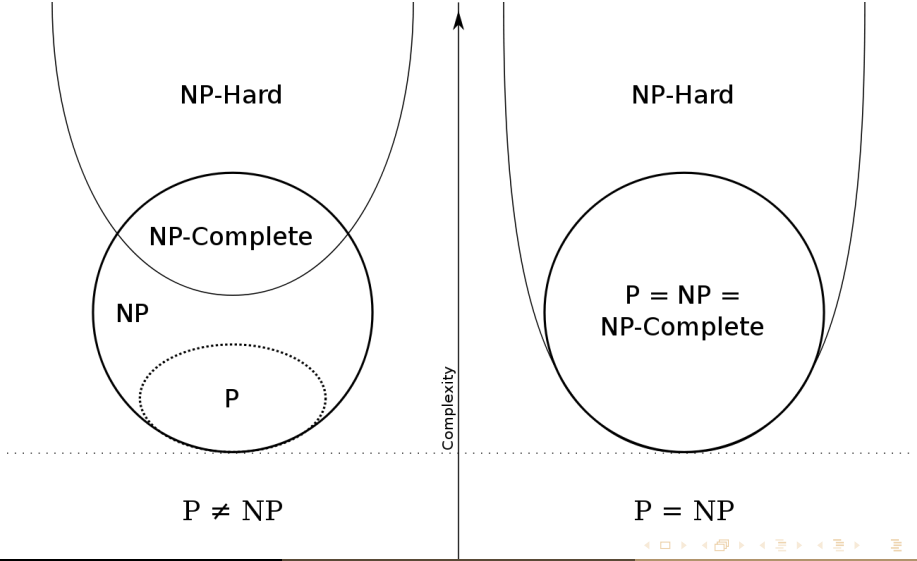
- Determinar a sequência de DNA que melhor se assemelha a um fragmento de DNA;
- Determinar procedimentos eficientes para predição de estrutura de proteínas;
- Determinar se uma afirmação matemática possui uma prova curta;
- Etc...

## P vs. NP

A partir destas descobertas, grande parte dos cientistas da computação passou a acreditar que P$\neq$NP.

Provar isto se tornou a questão mais importante da ciência da computação e uma das mais importantes da matemática.

# Problemas Intratáveis



## P vs. NP – Um dos Problemas do Milênio

O *Clay Mathematics Institute* elencou 7 problemas matemáticos e oferece um prêmio de um milhão de dólares para quem resolver um deles.

Provar que P=NP ou P!=NP é um dos 7 **Problemas do Milênio** desde o ano 2000[a]:

- P vs. NP;
- A conjectura de Poincaré (resolvido por Grigori Perelman em 2006);
- A conjectura de Hodge;
- A hipótese de Riemann;
- A existência de Yang-Mills e a falha na massa;
- A existência e suavidade de Navier-Stokes;
- A conjectura de Birch e Swinnerton-Dyer.

[a] Veja uma descrição informal em `http://goo.gl/wdWzzH`

## E Se P = NP?

"O que ganharíamos com P=NP faria com que a Internet inteira parecesse apenas um rodapé na história'.'
– Fortnow, L. 2009

## E Se P = NP?

Várias tarefas se tornariam triviais:

- Transporte de pessoas e produtos mais rápido e mais barato;
- Indústrias produzindo mais rápido e mais barato;
- Traduções automáticas;
- Reconhecimento de visão;
- Compreensão de linguagens;
- Previsão do tempo, terremotos e tsunamis;
- Provas curtas para teoremas matemáticos

## E Se P = NP?

Adeus criptografia!

A criptografia se baseia em problemas difíceis de serem resolvidos, como a fatoração de números muito grandes em números primos.

Portanto, é impossível de quebrar, a não ser que P=NP e fatoração seja um problema trivial...

## E Se P != NP?

Se for provado que P!=NP, não teremos os benefícios computacionais de P=NP.

Porém, ainda assim teríamos avanços na teoria da computação e uma direção para pesquisas futuras.

Se soubermos que um problema é intratável, não tentaremos resolvê-lo de maneira eficiente, ao invés disso, tentaremos soluções aproximadas ou parciais, com técnicas **apropriadas**.

Podemos utilizar a dificuldade em resolver problemas a nosso favor – como na criptografia.

## E Se P != NP?

Como dito anteriormente, os problemas de interesse são NP-Completos.

Ainda precisamos tentar resolvê-los, mesmo sem os benefícios de P=NP.

Mesmo não sendo totalmente eficientes, existem "bons" algoritmos para problemas industriais, matemáticos, computacionais, etc..

Aparecem aí a **Otimização Combinatória** e a **Pesquisa Operacional**.

## P vs. NP Atualmente

Existem 116 provas sobre P vs. NP, registradas pelo P vs. NP Page.

Em 2016 tivemos seis tentativas (igual:4, diferente:2).

De 26 de setembro de 2016 em diante, nenhuma atualização.

# Problemas Intratáveis

## Prof. Sóstenes Luiz Soares Lins (UFPE)

"Este trabalho permanece não publicado: continuo pesquisando que problemas de decisão são a ele redutíveis. No período 2008-2010, provei que MAX-CUT era um tal problema. Isto implicaria P=NP. Escrevi 52 versões da prova do resultado até descobrir, um erro em 10/01/2010."

"Numa época de ciência descartável (focada em número de publicações e estatísticas estranhas, dissociadas do conteúdo), orgulho-me de não transigir e ao menos tentar descobrir algo perene e importante."

## P vs. NP Atualmente

Um artigo de 2002 realizou uma pesquisa/entrevista com 100 dos maiores teóricos da computação do mundo – a intenção era obter um registro objetivo de opiniões subjetivas.

Dez anos depois (2012), a pesquisa/entrevista foi refeita, desta vez, com 200 participantes em potencial – 152 responderam.

Sete anos depois (2019), a terceira edição contou com 124 teóricos da computação, e uma pesquisa via SurveyMonkey.

## 2019

P = NP?

- Sim (12%);
- Não (88%) – somente *experts* vai a 99%.

*"NÃO. Pessoas que pensam P = NP são como pessoas que pensam que Elvis ainda está vivo. [Talvez Elvis prove P = NP.]"*
*– Lance Fortnow, Georgia Tech*

## Quando solucionaremos P vs. NP?

- Antes de 2100 (66%);
- Depois de 2100 (34%).

*"Eu espero que nos próximos 50 anos. Eu realmente gostaria de ver isso em minha vida, pois será um grande avanço, seja qual for a resposta. E terá impacto em outras grandes questões na complexidade computacional."*
*– Benoit Razet, Bucknell University*

## Quando solucionaremos P vs. NP?

*"O problema P =? NP nunca será resolvido. Estou pessimista de que alguém jamais provará P! = NP, mas estou otimista de que algum dia iremos provar que não há prova de que P != NP."*
*–William Hoza, The University of Texas at Austin*

## Qual será o método utilizado na prova?

- Novas técnicas (65 pessoas);
- Técnicas conhecidas (5 pessoas);
- Um algoritmo (6 pessoas);
- Milagre (1 pessoa);
- "Não saberemos até que seja resolvido";
- "Não vou te dizer";
- "Os aliens nos dirão";
- "Livros já amarelados, incluindo aqueles que sequer foram escritos ainda".

# P =? NP Poll

**Se alguém provar P=NP, haverá um grande efeito na computação prática?**

- Sim (58%);
- Não (42%).

*"Embora seja possível que a solução seja ineficaz, as consequências de um P = NP totalmente eficaz seriam enormes. Pode levar à imortalidade humana em 5 anos, ou se mantido em segredo por um grupo em busca de poder, o governo mundial em 2 anos."*
*–Dmytro Taranovsky, MIT*

*"O impacto prático viria não do resultado em si, mas das novas idéias necessárias para alcançá-lo."*
*–Scott Aaronson, MIT*

# P =? NP Poll

**Se alguém provar P!=NP, haverá um grande efeito na computação prática?**

- Sim (19%);
- Não (81%).

*"Já colocamos nossa fé em P != NP."*
*–Hal Gabow, University of Colorado*

# P =? NP Poll

## Algum outro comentário?

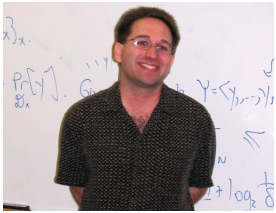
*"Sem provas, o melhor que temos são as opiniões de especialistas, incluindo diferenças de opinião."*
*–Dmytro Taranovsky, MIT*

## Algum outro comentário?

*"Mal posso esperar que as pessoas percebam que P =? NP é apenas uma questão sobre a existência de um algoritmo de tempo polinomial, não sobre o conhecimento de um."*
*–Donald Knuth, Stanford*

*"Espero que todos os matemáticos gostem das últimas décadas em que os matemáticos equipados com computador são significativamente mais fortes na prova de teoremas matemáticos do que os computadores sozinhos. Isso vai acabar logo. Em 20 anos, nosso papel será apertar o botão e ver a prova, ou, senão, concluir que se um computador não pode provar o teorema, então é impossível para os humanos sequer começarem."*
*–Bogdan Grechuk, University of Leicester*

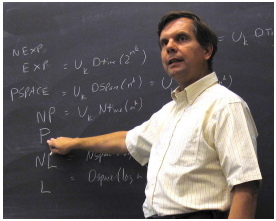
# P =? NP Poll



**Scott Aaronson, MIT**

*"Acredito em $P \neq NP$ basicamente pelo mesmo motivo que acho que não serei devorado amanhã por um sagui robótico de 150 metros de altura de Vênus, apesar de minha falta de provas em ambos os casos."*

# P =? NP Poll




**Eric Allender, Rutgers University**

*"P $\neq$ NP será resolvido em 25 anos, embora esta estimativa seja completamente sem sentido, é claro.*

*Técnicas: Se eu soubesse, não contaria a você."*

# P =? NP Poll

**Stephen Cook, Universidade de Toronto**

*"P $\neq$ NP não será resolvido nos próximos 20 anos e precisará de novas técnicas."*

# P =? NP Poll

**Lance Fortnow, Georgia Tech**

*"P $\neq$ NP. Isso será resolvido em um tempo imprevisivelmente longo. Se eu soubesse que tipo de técnica seria usada, não diria."*

# P =? NP Poll

**Richard Karp**

*"Eu acredito intuitivamente $P \neq NP$, mas é apenas uma intuição."*

# P =? NP Poll

## Michael Sipser, MIT

*"Até onde eu sei, não mudou muito matematicamente relevante para P vs. NP desde 2002, então não tenho nada a acrescentar à minha resposta anterior ao seu questionário.*

*Use minha resposta anterior; no entanto, atualize-o para que eu preveja que será resolvido daqui a 25 anos, não a partir de 2002.*

*(...) E continuarei com minha previsão anterior de que a resolução será uma prova de que $P \neq NP$."*

## Conclusão

"(...) my first reaction was the article could be written in two words: Still open." - Fortnow, L. 2009

## Referências


- Clay Mathematics Institute. The Millenium Problems. Disponível em: http://www.claymath.org/millennium/P_vs_NP/. Acessado em 21 de junho de 2023.
- Fortnow, L. 2009. The Status of the P Versus NP Problem. Communications of the ACM, Vol. 52 No. 9, Pages 78-86.
- The P vs. NP Page. Disponível em: http://www.win.tue.nl/~gwoegi/P-versus-NP.htm. Acessado em 21 de junho de 2023.
- Turing, A. On computable numbers, with an application to the Etscheidungs problem. Proceedings of the London Mathematical Society 42 (1936), 230-265.
- Gasarch, W. I. (2002). SIGACT News Complexity Theory Column 36. SIGACT News, 25.
- Gasarch, W. I. (2012). Guest column: The second P=? NP poll. ACM SIGACT News, 43(2), 53-77.
- Gasarch, W. I. (2019). Guest column: The third P=? NP poll. ACM SIGACT News, 50(1), 38-59.

# Dúvidas?