Classes de Complexidade

Entendendo problemas fáceis e difíceis

Prof. Me. Lucas Bruzzone

Aula 14

Uma Pergunta Simples

Por que alguns problemas são fáceis de resolver no computador e outros parecem impossíveis?

Dois Problemas, Duas Realidades

Situação 1: Encontrar seu nome numa lista

- 1 milhão de nomes
- Seu computador acha em menos de 1 segundo
- Problema FÁCIL

Situação 2: Planejar rota de entrega

- 20 endereços para visitar
- Qual a ordem mais rápida?
- Computador mais potente do mundo levaria ANOS
- Problema DIFÍCIL

Por que essa diferença?

Exemplos do Cotidiano

Problemas FÁCEIS para computadores:

- Ordenar uma lista de músicas
- Calcular o menor caminho no GPS
- Procurar uma palavra no Google

Problemas DIFÍCEIS para computadores:

- Montar a melhor escala de trabalho
- Otimizar rotas de entrega (Amazon, Uber Eats)
- Resolver Sudoku difícil
- Quebrar senhas criptografadas

O Mistério de US$ 1 Milhão

Conferir uma resposta é tão difícil quanto encontrar a resposta?

Pense no Sudoku:

- Resolver do zero: pode levar horas
- Conferir se está certo: leva minutos

Pense em uma senha:

- Descobrir a senha: impossível (milhões de tentativas)
- Conferir se está certa: instantâneo

Esta pergunta vale US$ 1.000.000 e ninguém sabe a resposta!

Classe P: Os Problemas Rápidos

P = Problemas que seu computador resolve rapidamente

O que significa "rapidamente"?

- O tempo não cresce absurdamente quando o problema aumenta
- Dobrou o tamanho? Tempo não multiplica por milhão

Pense assim:

- 10 elementos $\rightarrow$ 1 segundo
- 100 elementos $\rightarrow$ 10 segundos (não 1 milhão!)
- 1000 elementos $\rightarrow$ 100 segundos (não impossível!)

Exemplos da Classe P

1. **Ordenar uma lista**
   - Ordenar 1.000 nomes: menos de 1 segundo
   - Ordenar 1.000.000 nomes: alguns segundos

2. **GPS (menor caminho)**
   - Achar melhor rota entre 2 pontos
   - Mesmo com milhares de ruas

3. **Buscar no Google**
   - Procurar palavra em bilhões de páginas
   - Resultado em milissegundos

4. **Verificar se número é primo**
   - "17 é primo?" $\rightarrow$ rápido
   - Mesmo números gigantes!

Por Que P é Importante?

Se seu problema está em P, você tem solução!

Significa que:

- Existe algoritmo eficiente
- Funciona na prática
- Você pode implementar e usar

Exemplos reais:

- Apps de GPS funcionam porque achar rota está em P
- Google funciona porque busca está em P
- Spotify ordena suas playlists porque ordenação está em P

Classe NP: Fácil de Conferir

NP = Problemas fáceis de conferir, mas difíceis de resolver

Pense assim:

- Alguém te dá uma resposta
- Você consegue confirmar rapidamente se está certa
- Mas encontrar essa resposta do zero? Muito difícil!

NP NÃO significa "impossível"!

- Significa: "consigo conferir rapidamente"

Exemplo Prático: Sudoku

RESOLVER do zero:

- Tentar vários números
- Voltar atrás quando erra
- Pode levar muito tempo

CONFERIR se está certo:

- Olhar cada linha: tem 1 a 9?
- Olhar cada coluna: tem 1 a 9?
- Olhar cada quadrado: tem 1 a 9?
- Pronto! Rápido!

Conferir é MUITO mais fácil que resolver!

Mais Exemplos de NP

1. **Senha do celular**
   - Descobrir: testar milhões de combinações
   - Conferir: digitar e ver se abre (instantâneo)

2. **Mochila de viagem**
   - Escolher itens que cabem e valem mais: difícil
   - Conferir se uma escolha funciona: fácil (somar peso e valor)

3. **Escala de trabalho**
   - Montar escala que atenda todas regras: complicado
   - Conferir se uma escala funciona: fácil (ver se tem conflitos)

4. **Quebra-cabeça**
   - Montar: trabalhoso
   - Ver se está certo: rápido

P e NP: Qual a Relação?

Todo problema em P também está em NP

Por quê?

- Se você consegue resolver rápido
- Então também consegue conferir rápido
- É só resolver do zero e comparar!

Mas será que NP = P?

Em outras palavras:

- Todo problema fácil de conferir também é fácil de resolver?
- Ninguém sabe!
- Esta é a pergunta do milhão!

NP-Completo: Os Chefões

NP-Completo = Os problemas MAIS DIFÍCEIS de NP

Pense neles como:

- Os "chefões" dos problemas difíceis
- Se você resolver um, resolve TODOS os outros
- São todos igualmente difíceis

Por que isso importa?

- Se seu problema é NP-Completo, não perca tempo
- Não existe solução rápida conhecida
- Use estratégias alternativas (aproximações)

Problemas NP-Completos Famosos

1. **Caixeiro Viajante**
   - 20 cidades para visitar
   - Qual a ordem mais rápida?
   - Usado por: Uber Eats, iFood, correios

2. **Mochila (versão difícil)**
   - Escolher itens que maximizam valor
   - Sem passar do peso limite

3. **Coloração de mapas**
   - Colorir mapa com 3 cores
   - Países vizinhos não podem ter mesma cor

4. **Escala de horários**
   - Montar grade de aulas sem conflitos
   - Usado por: universidades, hospitais

Por Que NP-Completos São Especiais?

Todos têm a mesma dificuldade!

Significa que:

- Caixeiro viajante é tão difícil quanto Sudoku
- Sudoku é tão difícil quanto mochila
- Mochila é tão difícil quanto coloração

Consequência prática:

- Se alguém achar solução rápida para UM
- Automaticamente existe solução rápida para TODOS
- Por isso são chamados de "completos"

P = NP? A Pergunta do Milhão

Cenário 1: Se P = NP

- Todo problema fácil de conferir também é fácil de resolver
- Sudoku, senhas, rotas: tudo seria rápido
- Criptografia seria quebrada
- Revolução total na computação

Cenário 2: Se P != NP

- Existem problemas impossíveis de resolver rápido
- Conferir é fundamentalmente mais fácil que resolver
- Criptografia continua segura
- Precisamos de aproximações para problemas difíceis

99% dos cientistas acreditam que P != NP

Por Que Acreditamos que P != NP?

Intuição:

- Resolver parece sempre mais difícil que conferir
- Décadas de pesquisa sem achar algoritmos rápidos
- Se P = NP, já teríamos descoberto

Mas não temos prova!

- Ninguém conseguiu provar que P != NP
- Também ninguém provou que P = NP
- É um dos maiores mistérios da ciência
- Prêmio de US$ 1.000.000 para quem provar

Como Empresas Lidam com NP-Completos?

Se o problema é difícil demais, o que fazer?

1. **Aproximações**
   - Achar uma solução "boa o suficiente"
   - Exemplo: rota do iFood não é perfeita, mas é rápida
2. **Heurísticas (macetes)**
   - Regras práticas que funcionam bem
   - Não é ótimo, mas resolve
3. **Limitar o tamanho**
   - Para 10 cidades, até dá para calcular tudo
   - Mas não para 1000 cidades
4. **Casos especiais**
   - Versão geral é difícil, mas variação pode ser fácil

Exemplos Reais

iFood / Uber Eats:

- Problema: rota perfeita para 50 entregas (NP-Completo)
- Solução: algoritmo de aproximação (90% do ótimo)
- Resultado: rápido e funciona bem

Google Maps:

- Problema: rota com várias paradas (NP-Completo)
- Solução: aproximação + casos especiais
- Resultado: rota boa em segundos

Netflix:

- Problema: recomendar melhor combinação (difícil)
- Solução: heurísticas baseadas em padrões
- Resultado: recomendações boas, não perfeitas

O Que Você Deve Saber

1. **P** = problemas que o computador resolve rápido
2. **NP** = problemas fáceis de conferir, mas não necessariamente fáceis de resolver
3. **NP-Completo** = os problemas mais difíceis de NP
4. **P = NP?** = ninguém sabe! (US$ 1.000.000)
5. **Na prática:** se é NP-Completo, use aproximações

Perguntas para Pensar

1. Por que é útil saber se um problema é NP-Completo?
2. O que aconteceria se amanhã alguém provasse P = NP?
3. Você consegue pensar em outros problemas do dia a dia que são fáceis de conferir mas difíceis de resolver?
4. Por que empresas usam aproximações em vez de esperar a solução perfeita?

Dúvidas?

Obrigado!