

Universidade Federal de Ouro Preto

PCC104 - Projeto e Análise de Algoritmos

Problemas P, NP e NP-Completo

Prof. Rodrigo Silva

November 8, 2021

Natália Fernanda de Castro Meira

1 Leitura Recomendada

- Seção 11.3 - Introduction to the Design and Analysis of Algorithms (3rd Edition) - Anany Levitin

2 Atividades ([discussões na aula 08/11/2021 e 10/11/2021](#))

1. O que significa dizer que um algoritmo resolve um problema em tempo polinomial?

Significa que existe um algoritmo determinístico que executa em tempo polinomial e resolve o problema. Um algoritmo resolve um problema em tempo polinomial se a sua eficiência de pior caso pertence a $O(p(n))$, onde $p(n)$ é um polinômio do tamanho da entrada do problema n .

2. Que tipo de problemas considera-se tratável?

Problemas que podem ser resolvidos em tempo polinomial.

3. Que tipo de problema considera-se intratável?

Problemas que possuem solução em tempo exponencial.

4. Em ciência da computação, o que é o conjunto ou classe de problemas P?

Classe de problemas de decisão que são resolvidos em tempo polinomial.

5. Como podemos provar que um problema pertence a classe P?

Se existe um algoritmo determinístico que executa em tempo polinomial e resolve o problema, e se sua eficiência de pior caso pertence a $O(p(n))$, onde $p(n)$ é um polinômio do tamanho da entrada do problema n .

6. O que é um problema decidível? E um problema indecidível?

Problemas decidíveis são problemas que podem ser resolvidos por um algoritmo. Problemas indecidíveis são problemas de decisão que não podem ser resolvidos por nenhum algoritmo.

7. De forma geral, o que é um algoritmo determinístico?

Um algoritmo é determinístico se, para uma mesma entrada, sempre retorna a mesma saída.

8. De forma geral, o que é um algoritmo não determinístico?

Um algoritmo não-determinístico (estocástico) é aquele que, para uma mesma entrada, pode apresentar saídas diferentes.

9. Em ciência da computação, o que é o conjunto ou classe de problemas NP?

A classe de problemas NP (não-determinístico polinomial) são problemas de decisão executados em tempo polinomial que, para uma mesma entrada, pode fornecer saídas diferentes.

É executado como um procedimento em dois estágios: um estágio não determinístico (geração) e um estágio determinístico (verificação).

Ou seja, dizemos que um algoritmo não determinístico resolve um problema de decisão se, e somente se, para cada instância **sim** do problema, ele retorna **sim** em alguma execução.

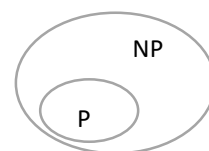
10. O que é um algoritmo polinomial não determinístico?

Explicação na questão anterior.

11. Explique por quê $P \subseteq NP$?

Pois satisfaz as condições:

- Para um problema P podemos **gerar** uma solução tem tempo polinomial, e;
- Para um problema P podemos **verificar** uma solução tem tempo polinomial.



12. Por quê saber se $P = NP$ é interessante?

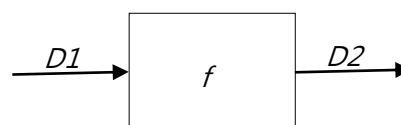
Porque se $P = NP$, então teríamos algoritmos eficientes para vários problemas, ou seja, vários problemas intratáveis seriam tratáveis.

13. Como provamos que um problema é NP-Completo?

Um problema NP completo:

- Pertence a classe NP;
- É redutível polinomialmente.

Um polinômio $D1$ é redutível a $D2$ se existe uma função f que transforma qualquer instância de $D1$ em uma instância de $D2$, e f é executado em tempo polinomial.



14. Como provamos que um problema é NP-Completo quando já conhecemos algum problema NP-Completo?

Para mostrar que D_x é NP-completo dado que D_y é NP-completo:

- Mostrar que D_x está em NP (construir e avaliar uma solução em tempo polinomial);
- Demonstrar que qualquer problema em NP é redutível à D_x polinomialmente, por exemplo, sabendo que D_y é NP-completo;
 - Mostrar que instâncias de D_y podem ser mapeadas para D_x em tempo polinomial;
 - Mapeamento: $D_x: \text{sim} \leftrightarrow D_y: \text{sim}$.

15. O que significaria resolver ao problema NP-Completo em $O(n^5)$?

Significaria que, ao transformar uma instância D_y para uma instância D_x de custo $O(n^5)$, somaria-se o custo da transformação ao custo total. Assim, a melhor hipótese é custo $O(n^5)$ e, esse custo piora, dependendo da contribuição da transformação.

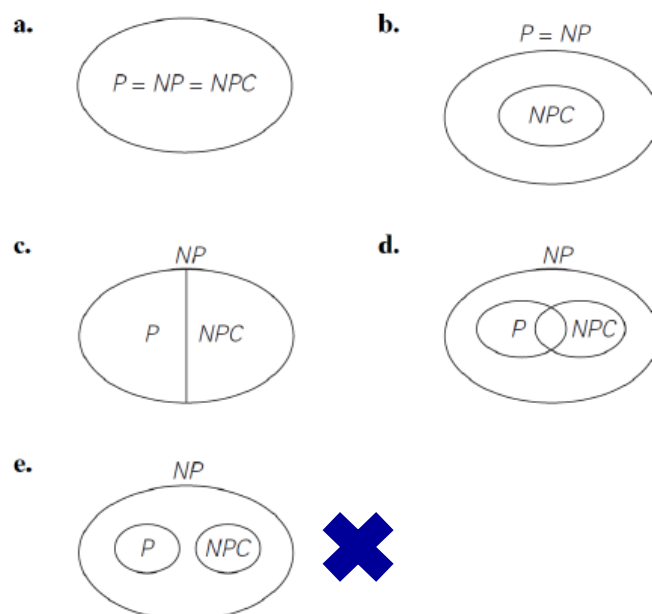
16. Um algoritmo que faz um número polinomial de chamadas a um procedimento que executa em tempo polinomial pode ter complexidade exponencial? Explique.

Sim. (caderno)

Fonte:

<<https://www.evernote.com/shard/s370/client/snv?noteGuid=9b61bfc9-00aa-5a8d-d792-cb1bd217140e¬eKey=7a3f69801481b9642f4de02f6b5d3e0f&sn=https%3A%2F%2Fwww.evernote.com%2Fshard%2Fs370%2Fsh%2F9b61bfc9-00aa-5a8d-d792-cb1bd217140e%2F7a3f69801481b9642f4de02f6b5d3e0f&title=Problemas%2BP%252C%2BNP%2Be%2BNP-completo>>

17. Qual dos diagramas abaixo não contradiz o estado corrente do nosso conhecimento sobre as classes de problemas P, NP e NP-Completo.



(caderno)

18. Mostre que o Problema do Conjunto independente é um problema NP-Completo utilizando a redução entre problemas, considerando o 3-SAT como problema base.

(caderno)