

NP-Completeness

Breno Naodi Kusunoki
`brenokusunoki@gmail.com`

Luiz Guilherme Castilho Martins
`luizgui@gmail.com`

Londrina, 20 de agosto de 2012

Sumário

1	Problemas com tempo polinomial	3
2	Problemas polinomiais não determinísticos	3
3	Problemas NP-Difícil	3
4	Problemas NP-Completo	3
5	Redutibilidade	4
6	Transferência de cotas	4
7	Exemplos	4
7.1	O Problema da Coloração de Grafos	4
7.2	Exemplo 2	4

1 Problemas com tempo polinomial

Problemas com tempo polinomial ou apenas problemas **P**. São problemas comprovadamente possíveis de serem resolvidos deterministicamente em tempo polinomial, isto é, esses problemas podem ser resolvidos no tempo $O(n^k)$ para alguma constante k e n o tamanho da entrada [1].

2 Problemas polinomiais não determinísticos

Problemas polinomiais não determinísticos ou apenas problemas **NP** é uma classe de problemas que não se conseguiu até este momento criar algoritmos determinísticos que os resolvam em tempo satisfatório [2].

Para esse tipo de problemas utiliza-se algoritmos não determinísticos como parte da solução e cria-se mecanismos para fazer a validação da solução "encontrada". Essa validação deve ter custo polinomial para que faça sentido o uso desta abordagem.

Podemos definir de forma mais informal que algoritmos não determinísticos são algoritmos que basicamente "escolhem" uma solução entre todas as possíveis soluções, sem nenhum critério de escolha. Essa "escolha" por sua vez tem custo $O(1)$ uma vez que não é realizado nenhum esforço computacional.

Pode-se acrescentar também que todos os problemas em **P** também estão em **NP**, ou seja o conjunto de problemas **P** está contido em **NP**, isso se dá pelo fato de que podemos resolver qualquer problema **P** em tempo polinomial [1].

3 Problemas NP-Difícil

Problemas NP-Difícil são problemas que podem ser redutíveis em tempo polinomial a todos os problemas NP porém esse problema não está contido em NP.

4 Problemas NP-Completo

Problemas NP-Completo são problemas que são NP e também é tão "difícil" quanto todos os outros problemas NP, e isso deve ser provado matematicamente assim como o *Teorema de Cook* com o *Problema SAT*, uma

vez que não é possível prová-los um-a-um devido a grande quantidade de problemas NP.

5 Redutibilidade

Redutibilidade é utilizado na prova de NP-Completeness. Reduzir um problema significa dizer que dado dois problemas **A** e **B**, existe uma função com tempo polinomial que transforma qualquer instância de **A** em instância de **B**, existindo também a condicional *se somente se* sobre a resposta, ou seja, se a resposta de **A** é "sim" a de **B** também é "sim".

6 Transferência de cotas

Transferência de cotas se dá quando há uma redução de um problema em outro, dessa forma existe uma transferência de cota inerente da redução.

Dada uma redução polinomial de **X** a **Y** ou seja $X \leq_P Y$ e o algoritmo de redução f de tempo polinomial. Sabendo que a cota superior de **Y** é $O(n \lg n)$ existirá uma transferência da cota superior de **Y** para a cota inferior de **X** acrescido do tempo do algoritmo de redução ou seja o a cota inferior de **X** será $O(n \lg n + f)$

7 Exemplos

7.1 O Problema da Coloração de Grafos

Um grafo $G(V, E)$ pode ser "colorido" com k cores?

O problema da coloração de grafos consiste em determinar o número mínimo de cores necessárias para colorir um grafo de forma que não exista 2 vértices adjacentes com a mesma cor. Este número é chamado de número cromático.

7.2 Exemplo 2

Referências

- [1] C. LEISERSON, R. RIVEST, T. CORMEN, and C. STEIN, *ALGORITMOS - TEORIA E PRÁTICA: TRADUÇÃO DA 2ª EDIÇÃO AMERICANA*. CAMPUS.
- [2] M. GOODRICH and R. TAMASSIA, *Projeto de Algoritmos*. BOOKMAN COMPANHIA ED.
- [3] J. E. Hopcroft, J. D. Ullman, and A. V. Aho, *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
- [4] L. V. Toscani and P. A. S. Veloso, *Complexidade de Algoritmos Analise, Projeto e Metodos*. Porto Alegre, RS: Editora Sagra Luzzatto, 1 ed., 2009.
- [5] S. S. Skiena, *The Algorithm Design MANUAL*. Springer, 2 ed., 2010.
- [6] K. N. Richard Neapolitan, *Foundations of Algorithms Using C++ Pseudocode*. Jones and Bartlett, 2 ed., 1997.
- [7] U. Manber, *Introduction To Algorithms - A Creative Approach*. Addison-Wesley, 1 ed., 1989.