MCTA003-17 - ANÁLISE DE ALGORITMOS
CENTRO DE MATEMÁTICA, COMPUTAÇÃO E COGNIÇÃO
UNIVERSIDADE FEDERAL DO ABC
PROF. MAYCON SAMBINELLI

UFABC

Reduções e Complexidade de Computacional

ATENÇÃO!
A definição dos problemas citados nas questões está na página seguinte.

1. Defina formalmente: algoritmo eficiente, problema de decisão, problema de otimização, certificado (positivo), algoritmo verificador, classes P, NP, NP-completo e NP-difícil.

2. Informalmente, o que significa dizer que um problema é NP-completo?

3. É verdade que se reduzirmos um problema em P para um problema em NP, então P = NP? Justifique.

4. Seja $A \in$ NP-completo. Seja B um outro problema qualquer.
   (a) Diga tudo que podemos concluir se $A \preccurlyeq B$.
   (b) Diga tudo que podemos concluir se $A \preccurlyeq_{\text{poli}} B$.
   (c) Diga tudo que podemos concluir se $B \preccurlyeq_{\text{poli}} A$.

5. Prove que CN-SAT $\in$ NP.

6. Seja G um grafo. Uma *coloração* de G é uma função $c\colon V(G) \to \{1, 2, \ldots, k\}$, para algum inteiro $k \geq 1$, tal que $c(u) \neq c(v)$ para toda aresta $uv \in E(G)$. Ela recebe esse nome pois pode ser visualizada como uma atribuição de cores aos vértices (cada inteiro entre 1 e k é uma cor). Uma *3-coloração* é uma coloração em que $k = 3$. Considere o problema da 3-coloração: dado um grafo G, determinar se ele tem ou não uma 3-coloração. Mostre que o problema da 3-coloração está em NP.

7. Dado um grafo G com pesos nas arestas, prove que determinar se existe uma árvore geradora com peso no máximo k está em NP.

8. O problema CAMK consiste em, dado um grafo G, uma função $w\colon E(G) \to \mathbb{R}$ e um número real k, decidir se existe um caminho em G de custo total no máximo k. O problema CAM consiste em, dado um grafo G, uma função $w\colon E(G) \to \mathbb{R}$ e um vértice s, encontrar o menor caminho entre s e qualquer outro $v \in V(G)$. Mostre como reduzir o problema CAMK para o problema CAM. O que pode ser concluído sobre CAMK?

9. Mostre que se o TSPK pode ser resolvido em tempo polinomial, então o TSP também pode.

10. Assumindo que TSPK é NP-completo, prove que TSP é NP-difícil. Problemas de otimização nunca são NP-completos, pois eles não pertencem à NP, que só contém problemas de decisão. Porém, eles podem ser NP-difíceis. Para resolver esse exercício basta fazer uma redução polinomial nos mesmos moldes das feitas entre problemas de otimização.

11. Mostre que o problema SET COVER é NP-completo usando o problema VERTEX COVER, que é NP-completo.

12. Prove que BARRA pertence à P.

13. Mostre que o problema INDEPENDENT SET é NP-completo usando o problema VERTEX COVER, que é NP-completo.

14. Sejam A e B dois problemas na classe P, C um problema na classe NP e D e E dois problemas na classe NP-completo. Verdadeiro ou falso: se existe algoritmo com tempo $\Theta(n^2)$ que resolve B e A pode ser reduzido para B em tempo polinomial, então não existe algoritmo com tempo $\Theta(n)$ que resolve A. Justifique.

Definições

Dado um (di)grafo G, uma função $w\colon E(G) \to \mathbb{R}_+$, e um sub(di)grafo $H \subseteq G$, definimos

$$w(H) = \sum_{e \in E(H)} w(e).$$

Problema 1 *TSP*
ENTRADA: *grafo G, função $w\colon E(G) \to \mathbb{R}_+$ de custo das arestas.*
SAÍDA: $\min\{w(C)\colon C$ *é um ciclo gerador de* $G\}$.

Problema 2 *TSPk*
ENTRADA: *grafo G, função $w\colon E(G) \to \mathbb{R}_+$ de custo das arestas, valor k.*
SAÍDA: *<u>sim</u>, se existe um ciclo gerador C tal que $w(C) \leq k$; e <u>não</u>, caso contrário.*

Problema 3 *Set Cover*
ENTRADA: *conjunto $T = \{u_1, \ldots, u_n\}$ de n elementos, coleção S_1,...,S_m de subconjuntos de T ($S_i \subseteq T$ para todo i), inteiro k.*
SAÍDA: *<u>sim</u>, se existe uma coleção de no máximo k conjuntos S_i tal que a união deles é T; <u>não</u>, caso contrário.*

Problema 4 *Vertex Cover*
ENTRADA: *grafo G e inteiro ℓ.*
SAÍDA: *<u>sim</u>, se existe um subconjunto de vértices $S \subseteq V(G)$ tal que $|S| \leq \ell$ e, para toda aresta $uv \in E(G)$, vale que pelo menos um dentre u e v estão em S; <u>não</u>, caso contrário.*

Problema 5 *Independent Set*
ENTRADA: *grafo G e inteiro z.*
SAÍDA: *<u>sim</u>, se existe subconjunto de vértices $S \subseteq V(G)$ tal que $|S| \geq z$ e, para todo par $u, v \in S$, vale que $uv \notin E(G)$; <u>não</u>, caso contrário.*

Problema 6 *CN-SAT*
ENTRADA: *uma fórmula ϕ que é a conjunção de m cláusulas C_i, com cada cláusula sendo uma disjunção de literais, e cada literal sendo uma variável ou sua negação, tirada do conjunto $\{x_1, \ldots, x_n\}$ de variáveis. Em outras palavras, $\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ e $C_i = \ell_1 \vee \ell_2 \vee \cdots \vee \ell_{k_i}$, para $1 \leq i \leq m$, e $\ell_j \in \{x_1, \ldots, x_n\} \cup \{\overline{x_1}, \ldots, \overline{x_n}\}$, para $1 \leq j \leq k_i$.*
SAÍDA: *<u>sim</u>, se existe uma atribuição de valores lógicos às variáveis $x_1, x_2, \ldots, x_n$ tal que ϕ seja avaliada para verdadeiro.*

Problema 7 *BARRA*
ENTRADA: *inteiro n, um vetor $p[1..\ell]$, onde $n \leq \ell$ e $p[i] \in \mathbb{R}_+$ para todo $i = 1, \ldots, \ell$, e um inteiro k.*
SAÍDA: *sim, se é possível cortar uma barra de comprimento n em pedaços $b_1, b_2, \ldots, b_r$ tais que $\sum_{i=1}^{r} p[i] \geq k$; não, caso contrário.*