



Certifique-se de que
você tentou fazer por
um tempo antes de ver
as próximas páginas!

2. Defina formalmente a classe NP e a classe NP-completo. Informalmente, o que significa dizer que um problema é NP-completo?

Classe NP: problemas de decisão para os quais existe um algoritmo verificador de tempo polinomial.

Classe NP-completo: problemas de decisão para os quais todo problema em NP pode ser reduzido em tempo polinomial e que também estão em NP.

Informalmente, dizer que um problema é NP-completo significa dar um atestado de dificuldade para sua solução: a menos que $P=NP$, não há algoritmo eficiente que o resolva.

4. Seja A um problema NP-completo. Seja B um outro problema qualquer. Diga tudo que podemos concluir ao reduzir A para B . Diga tudo que podemos concluir ao reduzir B para A .

Seja A NP-completo e B qualquer.

Ao reduzir de A para B conseguimos mostrar que todo problema de NP reduz para B e portanto B é NP-difícil.

Se sabermos que $B \in \text{NP}$ (mas não sabemos), poderíamos concluir que B é NP-completo também.

Ao reduzir de B para A , estamos apenas criando uma forma (possivelmente bem ruim) de resolver B .

→ não podemos concluir que $B \in \text{NP}$!

6. Seja G um grafo. Uma *coloração* de G é uma função $c: V(G) \rightarrow \{1, 2, \dots, k\}$, 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.

3-coloração está em NP porque podemos criar um algoritmo verificador de tempo polinomial para ele: dados um grafo G e uma função $c: V(G) \rightarrow \{1, \dots, k\}$, podemos para cada vértice verificar se sua cor e a de seus vizinhos são diferentes em tempo polinomial (um único percurso no grafo todo) e também podemos verificar se $k=3$.

outra possibilidade é já dizer que $c: V(G) \rightarrow \{1, 2, 3\}$, restando apenas verificar a questão dos arestos.

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

Primeiro note que SETCOVER está em NP: dada uma coleção de subconjuntos de T , é possível em tempo polinomial verificar se a união deles é igual a T e se existem no máximo k deles.

Suponha seja $\langle G, l \rangle$ uma instância de VERTEXCOVER.

Vamos construir uma instância $\langle T, n, S, m, k \rangle$ para o SETCOVER da seguinte forma:

- $T = E(G)$
- $S_i = \{e : \text{a aresta } e \text{ incide no vértice } i\} \quad \forall i \in V(G)$
- $n = |E(G)|$
- $m = |V(G)|$
- $k = l$

① Suponha que $\langle G, l \rangle$ é Sim.

Então existe conjunto $X \subseteq V(G)$, com $|X| \leq l$, que cobre todas as arestas de G , isto é, toda aresta $uv \in E(G)$ tem $u \in X$ ou $v \in X$.

Para cada $i \in X$, tome os conjuntos S_i correspondentes.

Como X cobre as arestas, todo $uv \in T$ está em algum conjunto escolhido (S_u ou S_v).

Então a união dos conjuntos escolhidos (no máximo $l = k$) é T .

Logo, $\langle T, n, S, m, k \rangle$ é Sim.

② Suponha que $\langle T, n, S, m, k \rangle$ é Sim.

Então existem no máximo k conjuntos S_i que, quando unidos, resultem em T .

Construa um conjunto X de vértices i tais que S_i é um destes.

Como $T = E(G)$, para toda $uv \in E(G)$ temos $u \in X$ ou $v \in X$.

E como $|X| \leq k = l$, temos que $\langle G, l \rangle$ é Sim.

QED