Sistemas de Informação
UFPA

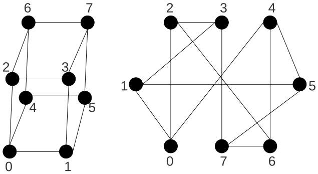
Grafos

Conceitos Básicos

26 de outubro de 2020

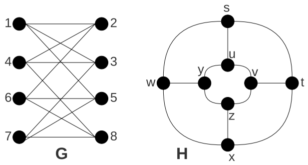
Isomorfismo

Os seguintes grafos são "iguais" ?



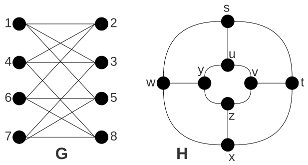
Isomorfismo

Os seguintes grafos são "iguais" ?



Isomorfismo

Os seguintes grafos são "iguais" ?

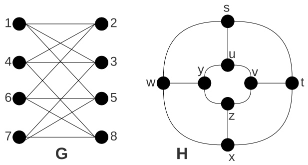


Precisamos encontrar uma função **Bijeção** $f : V_1 \rightarrow V_2$ sobre os vértices.

$$
\begin{array}{ll}
1 \rightarrow s & 5 \rightarrow w \\
2 \rightarrow t & 6 \rightarrow x \\
3 \rightarrow u & 7 \rightarrow y \\
4 \rightarrow v & 8 \rightarrow z
\end{array}
$$

Isomorfismo

Os seguintes grafos são "iguais" ?



Precisamos encontrar uma função **Bijeção** $f : V_1 \rightarrow V_2$ sobre os vértices.

$$\begin{array}{ll} 1 \rightarrow s & 5 \rightarrow w \\ 2 \rightarrow t & 6 \rightarrow x \\ 3 \rightarrow u & 7 \rightarrow y \\ 4 \rightarrow v & 8 \rightarrow z \end{array}$$

Observe que as **adjacências também devem ser preservadas**. $s = f(1)$ é adjacente somente a $t = f(2)$, $u = f(3)$ e $w = f(5)$.

Isomorfismo

Def.:

Sejam $G = (V_1, A_1)$ e $H = (V_2, A_2)$ dois grafos simples. G e H são isomorfos se existir uma função bijeção $f : V_1 \to V_2$ que preserva a estrutura (adjacências e não-adjacências).

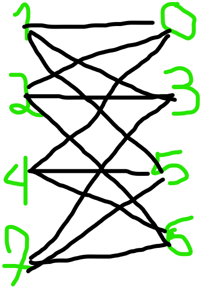
Isomorfismo

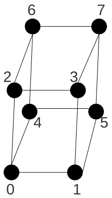
Def.:

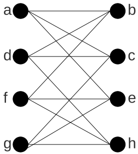
Sejam $G = (V_1, A_1)$ e $H = (V_2, A_2)$ dois grafos simples. G e H são isomorfos se existir uma função bijeção $f : V_1 \rightarrow V_2$ que preserva a estrutura (adjacências e não-adjacências).

Exercício

Dado os seguintes grafos, verifique se eles são isomorfos apresentado a função bijeção dos vértices que preserva a estrutura.







$V_1 = 0, 3, 5, 6$

$V_2 = 1, 2, 4, 7$

Generalizando o Isomorfismo

Sejam dois grafos $G = (V_1, A_1)$ e $H = (V_2, A_2)$ contendo ou não laços e/ou múltiplas arestas.

A bijeção de vértices $f : V_1 \to V_2$ preserva a estrutura se:

- O número de arestas entres todos os pares distintos de vértices u e v em G deve ser igual ao número de arestas entre suas imagens $f(u)$ e $f(v)$ em H.
- O número de laços em cada vértice x de G deve ser igual ao número de laços no vértice $f(x)$ de H.

Generalizando o Isomorfismo

Sejam dois grafos $G = (V_1, A_1)$ e $H = (V_2, A_2)$ contendo ou não laços e/ou múltiplas arestas.

A bijeção de vértices $f : V_1 \to V_2$ preserva a estrutura se:

- O número de arestas entres todos os pares distintos de vértices u e v em G deve ser igual ao número de arestas entre suas imagens $f(u)$ e $f(v)$ em H.
- O número de laços em cada vértice x de G deve ser igual ao número de laços no vértice $f(x)$ de H.

Def. Isomorfismo para grafos com/sem laços e/ou múltiplas arestas:
G e H são isomorfos se existir uma função bijeção $f : V_1 \to V_2$ que preserva a estrutura.

Generalizando o Isomorfismo

Teorema:

- Grafos isomorfos possuem a mesma sequência de graus

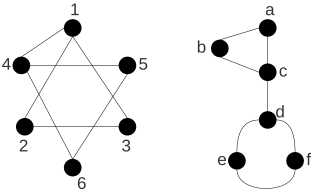
Generalizando o Isomorfismo

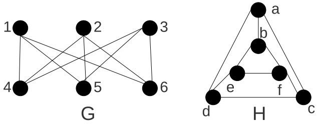
Teorema:

- Grafos isomorfos possuem a mesma sequência de graus
- Dois grafos não são isomorfos se um deles contém um subgrafo que não pertence ao outro

Exercício

Encontre o isomorfismo, se existir, nos seguintes pares de grafo:





O problema do isomorfismo de grafos

Dados dois grafos G e H com um grande número de vértices, é possível decidir se eles são isomórfos ?

O problema do isomorfismo de grafos

Dados dois grafos G e H com um grande número de vértices, é possível decidir se eles são isomórfos ?

Verificar se uma dada bijeção de vértices é um isomorfismo requer examinar todos os pares de vértices

O problema do isomorfismo de grafos

Dados dois grafos G e H com um grande número de vértices, é possível decidir se eles são isomórfos ?

Verificar se uma dada bijeção de vértices é um isomorfismo requer examinar todos os pares de vértices

Existem $n!$ possíveis bijeções de vértices
$\rightarrow$ força-bruta inviável para grandes grafos !!! :-(

Problema do Isomorfismo

Desenvolver um algoritmo polinomial para encontrar o isomorfismo ou provar que este algoritmo não existe.

Melhor Algoritmo para Grafo Isomorfismo

- O melhor algoritmo conhecido para Grafo Isomorfismo foi proposto em 2015:
  "Graph Isomorphism in Quasipolynomial Time", László Babai
  `https://arxiv.org/abs/1512.03547v1`

Melhor Algoritmo para Grafo Isomorfismo

- O melhor algoritmo conhecido para Grafo Isomorfismo foi proposto em 2015:
"Graph Isomorphism in Quasipolynomial Time", László Babai
`https://arxiv.org/abs/1512.03547v1`
- Tempo de execução: quasipolynomial $2^{(log n)^{O(1)}}$, para um número de vértices n
- Não se sabe se ele está em P ou NP-completo

Ainda Sobre Isomorfismo de Grafos

Trabalho

Fazer uma resenha de 1 página sobre os textos:

"Landmark Algorithm Breaks 30-Year Impasse" 2015
https://www.quantamagazine.org/algorithm-solves-graph-isomorphism-in-record-time-20151214/

"Graph Isomorphism Vanquished — Again" 2017
https://www.quantamagazine.org/graph-isomorphism-vanquished-again-20170114

Grafos Planares

Seja um grafo $G = (V, A)$ um grafo com uma representação R em um plano. R é plana quando não houver cruzamento de arestas.

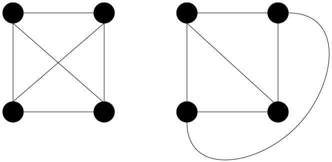
Grafos Planares

Seja um grafo $G = (V, A)$ um grafo com uma representação R em um plano. R é plana quando não houver cruzamento de arestas.

Grafo Planar

É um grafo que pode ser representado em um plano sem que haja cruzamento de arestas.

Ex.:



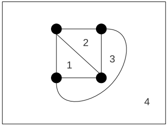
Grafos Planares

Seja um grafo planar $G = (V, A)$ com representação R em um plano P. R divide P em regiões (externa e limitadas).

Grafos Planares

Seja um grafo planar $G = (V, A)$ com representação R em um plano P. R divide P em regiões (externa e limitadas).

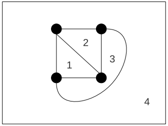
Ex.:



Grafos Planares

Seja um grafo planar $G = (V, A)$ com representação R em um plano P. R divide P em regiões (externa e limitadas).

Ex.:



Em um grafo planar, o número de vértices, o número de arestas e o número de regiões não são independentes.

Teorema (fórmula de Euler)

Seja um grafo simples planar $G = (V, A)$, n o número de vértices, m o número de arestas e r o número de regiões, $n + r = m + 2$.

Grafos Planares

Limite máximo para o número de arestas em um grafo planar (Corolário)

$$\text{Seja um grafo planar } G = (V, A),\ m \leq 3n - 6$$

O Corolário estabelece uma condição necessária, mas não suficiente para a planaridade.

Existem grafos que satisfazem a condição, mas não são planares.

Ex: $K_{3,3}$ tem $n = 3$ e $m = 9$ satisfazendo $9 = 3(6) - 6$, mas não é planar.

Uma API para Operações Básicas em Grafos Não Direcionados

Definindo uma API Java para operações básicas de grafos:

public class Grafo	
Grafo(int V)	Criar um grafo com V Vértices, sem arestas
int V()	Número de vértices
int A()	Número de arestas
void addAresta(int v, int w)	Adiciona a aresta v-w ao grafo
Iterable<integer> adj(int v)	Vértices adjacentes a 'v'
String toString()	representação string

O próximo passo é definir o tipo de representação do grafo.

Tarefas para Próxima Aula

- Estudar a implementação de listas encadeadas em Java;
- Estudar classes genéricas em Java;
- Estudar o laço do tipo *foreach* em Java.

Bibliografia

- LEISERSON , Charles E.; STEIN, C.; RIVEST, Ronald L., CORMEN, Thomas H. **Algoritmos: Teoria e Prática**, 1ª edição, Campus, 2002 (caps. 22 à 26);
- GROSS, Jonthan L., YELLEN, Jay. **Graph Theory and Its Applications**, Second Edition, Chapman and Hall/CRC, 2005.
- SEDGEWICK, Robert. **Algorithms in Java, Part 5: Graph Algorithms**, 3rd Edition, Addison-Wesley Professional, 2003;
- Goldbarg M.; Goldbarg E.; **Grafos: Conceitos, algoritmos e aplicações**, 1ª edição, Campus, 2012;