TEG

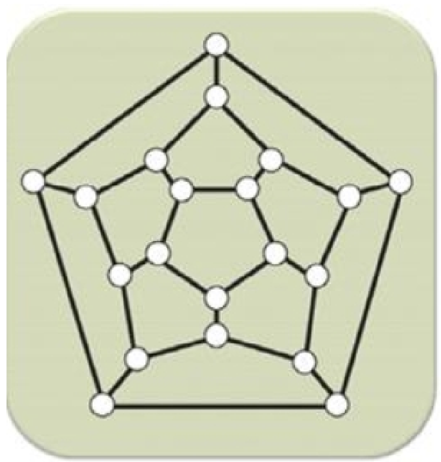
Gilmário B. Santos

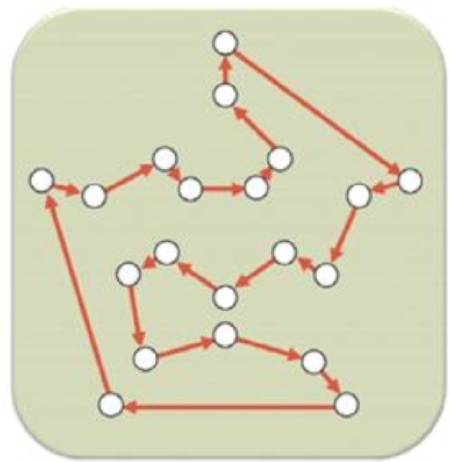
gilmario.santos@udesc.br

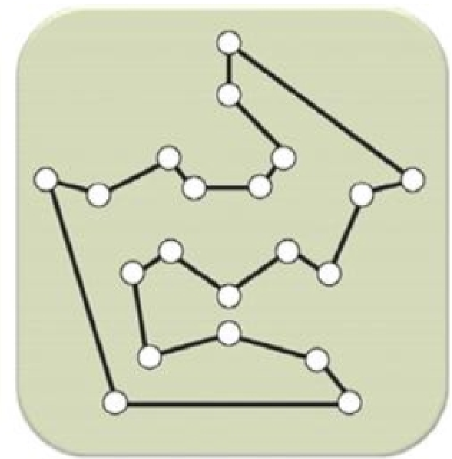
http://www.joinville.udesc.br/portal/pagina/gilmario

Grafos Hamiltonianos

Um grafo é Hamiltoniano se apresenta um ciclo que contenha todos os vértices do grafo sem repetições. Tal ciclo é dito “ciclo Hamiltoniano”.

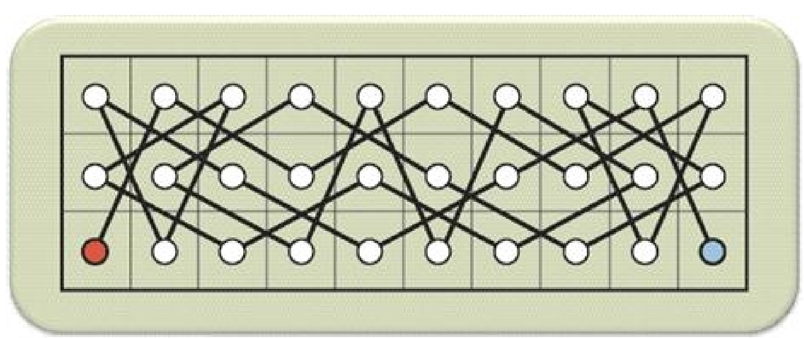

(1) Grafo que representa o jogo


(2) Uma solução do jogo


(3) Um ciclo hamiltoniano

Um grafo é dito semi-hamiltoniano se possui um caminho hamiltoniano (todo grafo hamiltoniano é semi-hamiltoniano). O grafo dos movimentos do cavalo (xadrez) é semi-hamiltoniano.

Não é exigida a presença de todas as arestas, porém, não há repetições de arestas.



Grafos Hamiltonianos

A literatura informa que não existe uma condição necessária e suficiente que caracterize, de maneira geral, um grafo G como Hamiltoniano.

Apesar disso há algumas situações que sinalizam um grafo hamiltoniano, exemplo: grafos completos com n>=3 vértices são hamiltonianos.

O Problema de Decisão associado a determinação de ciclos e caminhos hamiltonianos em grafos sem propriedades particulares é NP-Completo.

“Problemas NP-completos têm um status (de certa forma) desconhecido: ainda não foi descoberto nenhum algoritmo de tempo polinomial para um problema NP-Completo, nem ninguém ainda foi capaz de provar que não pode existir tal algoritmo para um problema NP-Completo.” Cormen.

Grafos Hamiltonianos

Um grafo é Hamiltoniano se apresenta um ciclo que contenha todos os vértices do grafo sem repetições. Tal ciclo é dito "ciclo Hamiltoniano".

Não existe uma condição necessária e suficiente que caracterize G como Hamiltoniano.

O Problema de Decisão associado à determinação de ciclos e caminhos hamiltonianos em grafos sem propriedades particulares é NP-Completo.

Problemas da classe NP-completo (nondeterministic polynomial) são muito difíceis de se resolver com métodos enumerativos pois o tempo de processamento é absolutamente inviável a não ser quando instâncias muito pequenas de problemas estão sendo resolvidas, já que o aumento do tempo do processamento é exponencial.

- Algoritmos exatos em ciência da computação e em pesquisa operacional, são algoritmos que sempre fornecem a solução ótima para um determinado problema. O exemplo mais emblemático de um algoritmo exato é um algoritmo que enumera todas as soluções possíveis, este tipo de algoritmo é chamado de enumerativo.

Grafos Hamiltonianos

Para contornar a ausência prática de algoritmos exatos ou enumerativos podem ser usadas heurísticas que em vários casos podem encontrar soluções suficientemente boas. Ou seja, soluções de boa relação custo (ou tempo de processamento)/benefício, considerando-se os tamanhos da instâncias.

Segundo o prof. Cid C. de Souza / IC–UNICAMP:

Heurísticas são algoritmos que geram soluções viáveis para quais não se pode dar garantias de qualidade. Ou seja, não se sabe o quão distante a solução gerada está de uma solução ótima (5%?, 10%?, 50%?, 100%?, ...).
Tipos de heurísticas:
Construtivas: normalmente adotam estratégias gulosas para construir as soluções. Tipicamente são aplicadas a problemas onde é relativamente fácil (mais trivial) obter uma solução viável.

Busca local: partem de uma solução inicial e, através de transformações bem definidas, visitam outras soluções até atingir um critério de parada pré-definido. Um exemplo clássico é o algoritmo Branch and Bound (algoritmo de Little, Marty, Sweeney & Karel)

Grafos Hamiltonianos

O clássico Problema do caixeiro viajante:

O problema do caixeiro viajante ou TSP (*Traveling Salesman Problem*), consiste em determinar um percurso em um roteiro de cidades, passando uma única vez em cada cidade e voltando à cidade de origem (**ciclo Hamiltoniano**), com **custo total mínimo**.

Pesquisa exaustiva pelo ciclo hamiltoniano de menor custo em $\mathbf{K_n}$:

- Um **grafo completo** de ordem n>=3 tem $1/2(n - 1)!$ ciclos de Hamilton candidatos a ciclos de peso mínimo;
- Mesmo para valores de n não muito grandes, este número continua a ser muito elevado. Ex.: n = 12 obtêm-se ½ * 11! = 19.958.400 ciclos hamiltonianos candidatos a ciclos de peso mínimo.

Ainda que se trate de um K_n, a pesquisa exaustiva ainda apresenta um custo computacional muito elevado, exigindo um esforço computacional que cresce como uma função fatorial na quantidade de vértices (n).

Alternativamente ao método de pesquisa exaustiva, é comum, na prática, a utilização de métodos heurísticos.

Fonte: https://core.ac.uk/download/pdf/15564607.pdf

Grafos Hamiltonianos

Heurística do Vizinho-mais-próximo (Prof. Cid C. de Souza / IC–UNICAMP):

- ▷ Exemplo 1: TSP em um grafo não orientado completo.

```
Vizinho-Mais-Próximo(n, d)      (* d: matriz de distâncias *)
    Para i = 1 até n faça visitado[i] ← Falso;
    visitado[1] ← Verdadeiro;
    ciclo ← {}, comp ← 0 e k ← 1;
    Para i = 1 até n − 1 faça
        j* ← argmin{d[k, j] : visitado[j] = Falso};
        visitado[j*] ← Verdadeiro;
        ciclo ← ciclo ∪ {(k, j*)};     comp ← comp + d[k, j*];
        k ← j*;
    fim-para
    ciclo ← ciclo ∪ {(k, 1)};     comp ← comp + d[k, 1];
Retorne comp.
```

- ▷ Complexidade: $O(n^2)$

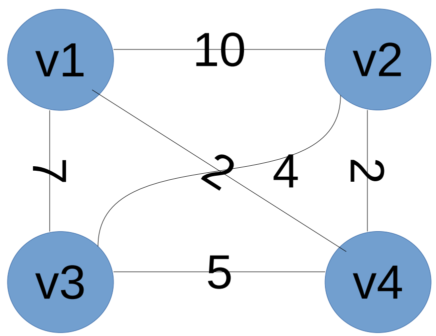
Simulando o algoritmo do vizinho mais próximo para um grafo completo ponderado

Heurística do Vizinho-mais-próximo (Prof. Cid C. de Souza / IC-UNICAMP):

▷ Exemplo 1: TSP em um grafo não orientado completo.

Vizinho-Mais-Próximo(n, d) (* d: matriz de distâncias *)
  Para $i = 1$ **até** n **faça** visitado[i] $\leftarrow$ Falso;
  visitado[1] $\leftarrow$ Verdadeiro;
  ciclo $\leftarrow \{\}$, comp $\leftarrow 0$ e $k \leftarrow 1$;
  Para $i = 1$ **até** $n - 1$ **faça**
    $j^* \leftarrow$ argmin$\{d[k, j]$: visitado[j] $=$ Falso$\}$;
    visitado[j^*] $\leftarrow$ Verdadeiro;
    ciclo $\leftarrow$ ciclo $\cup \{(k, j^*)\}$; comp $\leftarrow$ comp $+ d[k, j^*]$;
    $k \leftarrow j^*$;
  fim-para
  ciclo $\leftarrow$ ciclo $\cup \{(k, 1)\}$; comp $\leftarrow$ comp $+ d[k, 1]$;
Retorne comp.

Entenda a resposta de “argmin” como a aresta de menor peso que conecta o “k” a um vértice não visitado



Inicialização antes do 2º laço:

	v1	v2	v3	v4
Visitado:	T	F	F	F

Ciclo: vazio (custo do ciclo)

Comp=0

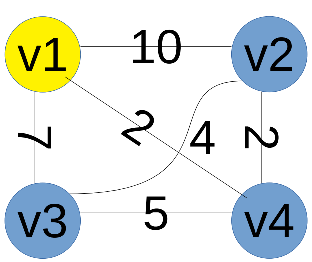
K= v1

Heurística do Vizinho-mais-próximo (Prof. Cid C. de Souza / IC–UNICAMP):

▷ Exemplo 1: TSP em um grafo não orientado completo.

```
Vizinho-Mais-Próximo(n, d)      (* d: matriz de distâncias *)
    Para i = 1 até n faça visitado[i] ← Falso;
    visitado[1] ← Verdadeiro;
    ciclo ← {}, comp ← 0 e k ← 1;
    Para i = 1 até n − 1 faça
        j* ← argmin{d[k, j] : visitado[j] = Falso};
        visitado[j*] ← Verdadeiro;
        ciclo ← ciclo ∪ {(k, j*)};     comp ← comp + d[k, j*];
        k ← j*;
    fim-para
    ciclo ← ciclo ∪ {(k, 1)};     comp ← comp + d[k, 1];
Retorne comp.
```

Entenda a resposta de "argmin" como a aresta de menor peso que conecta o "k" a um vértice não visitado



k=v1

j*=argmin{ d[k,j] : visitado[j]=falso}

j*=argmin{d(v1,v4);d(v1,v2);d(v1,v3)}=v4

(k,j*)=(v1,v4)

aresta de menor peso (2) que conecta k=v1 a um vértice j

i=1

	v1	v2	v3	v4
Visitado:	T	F	F	T

Ciclo: {(v1,v4)}

Comp=2

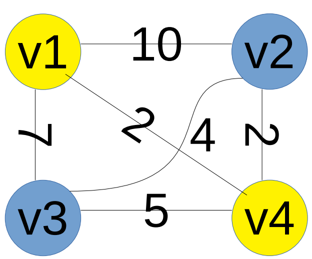
K= v4

Heurística do Vizinho-mais-próximo (Prof. Cid C. de Souza / IC–UNICAMP):

▷ Exemplo 1: TSP em um grafo não orientado completo.

`Vizinho-Mais-Próximo`(n, d) (* d: matriz de distâncias *)
- **Para** $i = 1$ **até** n **faça** visitado$[i] \leftarrow$ Falso;
- visitado$[1] \leftarrow$ Verdadeiro;
- ciclo $\leftarrow \{\}$, comp $\leftarrow 0$ e $k \leftarrow 1$;
- **Para** $i = 1$ **até** $n-1$ **faça**
  - $j^* \leftarrow$ argmin$\{d[k,j]$: visitado$[j] =$ Falso$\}$;
  - visitado$[j^*] \leftarrow$ Verdadeiro;
  - ciclo $\leftarrow$ ciclo $\cup \{(k, j^*)\}$; comp $\leftarrow$ comp $+ d[k, j^*]$;
  - $k \leftarrow j^*$;
- **fim-para**
- ciclo $\leftarrow$ ciclo $\cup \{(k, 1)\}$; comp $\leftarrow$ comp $+ d[k, 1]$;

Retorne comp.



k=v4
j*=argmin{ d[k,j] : visitado[j]=falso}

j*=argmin{d(v4,v2);d(v4,v3)}=v2

(k,j*)=(v4,v2)

aresta de menor peso (2) que conecta
k=v4 a um vértice j

i=2

	v1	v2	v3	v4
Visitado:	T	T	F	T

Ciclo: {(v1,v4),(v4,v2)}

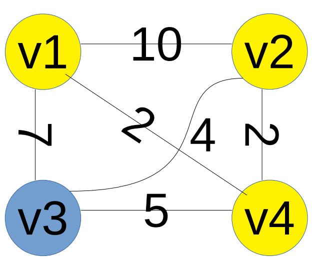
Comp=4

K= v2

Heurística do Vizinho-mais-próximo (Prof. Cid C. de Souza / IC−UNICAMP):

▷ Exemplo 1: TSP em um grafo não orientado completo.

Vizinho-Mais-Próximo(n, d) (* d: matriz de distâncias *)
Para $i = 1$ **até** n **faça** visitado$[i] \leftarrow$ Falso;
visitado$[1] \leftarrow$ Verdadeiro;
ciclo $\leftarrow \{\}$, comp $\leftarrow 0$ e $k \leftarrow 1$;
Para $i = 1$ **até** $n - 1$ **faça**
$j^* \leftarrow \text{argmin}\{d[k, j] \ : \ \text{visitado}[j] = \text{Falso}\}$;
visitado$[j^*] \leftarrow$ Verdadeiro;
ciclo $\leftarrow$ ciclo $\cup \{(k, j^*)\}$; comp $\leftarrow$ comp $+ d[k, j^*]$;
$k \leftarrow j^*$;
fim-para
ciclo $\leftarrow$ ciclo $\cup \{(k, 1)\}$; comp $\leftarrow$ comp $+ d[k, 1]$;
Retorne comp.



k=v2
j*=argmin{ d[k,j] : visitado[j]=falso}

j*=argmin{d(v2,v3)}=v3

(k,j*)=(v2,v3)

aresta de menor peso (4) que conecta
k=v2 a um vértice j

i=3

	v1	v2	v3	v4
Visitado:	T	T	T	T

Ciclo: {(v1,v4),(v4,v2),(v2,v3)}

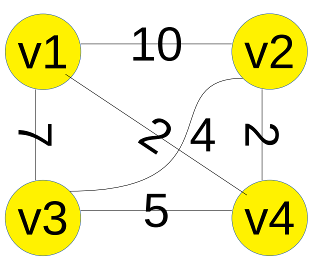
Comp=8

K= v3

Heurística do Vizinho-mais-próximo (Prof. Cid C. de Souza / IC–UNICAMP):

▷ Exemplo 1: TSP em um grafo não orientado completo.

```
Vizinho-Mais-Próximo(n, d)      (* d: matriz de distâncias *)
    Para i = 1 até n faça visitado[i] ← Falso;
    visitado[1] ← Verdadeiro;
    ciclo ← {}, comp ← 0 e k ← 1;
    Para i = 1 até n − 1 faça
        j* ← argmin{d[k, j] : visitado[j] = Falso};
        visitado[j*] ← Verdadeiro;
        ciclo ← ciclo ∪ {(k, j*)};      comp ← comp + d[k, j*];
        k ← j*;
    fim-para
    ciclo ← ciclo ∪ {(k, 1)};      comp ← comp + d[k, 1];
Retorne comp.
```



j*=argmin{d(v2,v3)}=v3

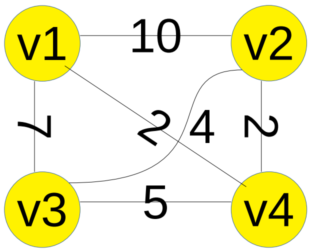
i=4

Fim do 2º laço

k=v3

Ciclo = ciclo U {(k,v1)}
= ciclo U {(v3,v1)}
={(v1,v4),(v4,v2),(v2,v3),(v3,v1)}

comp=comp+d(v3,v1)=8+7=15



Ciclo hamiltoniano: {(v1,v4),(v4,v2),(v2,v3),(v3,v1)}

Com custo total =15

Imagine um grafo completo com 5 vértices (A, B, C, D, E) e as seguintes arestas e pesos:

A - B: 1

A - C: 2

A - D: 3

A - E: 4

B - C: 5

B - D: 6

B - E: 7

C - D: 8

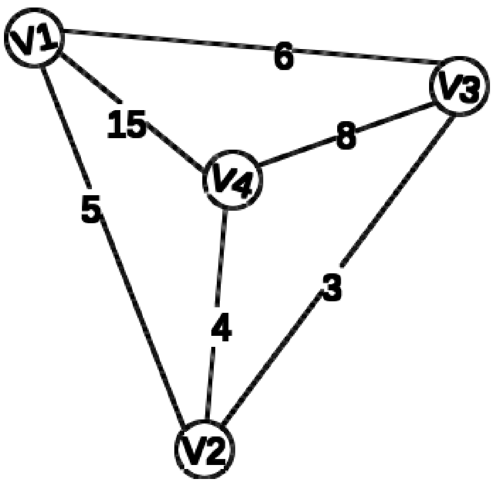
C - E: 9

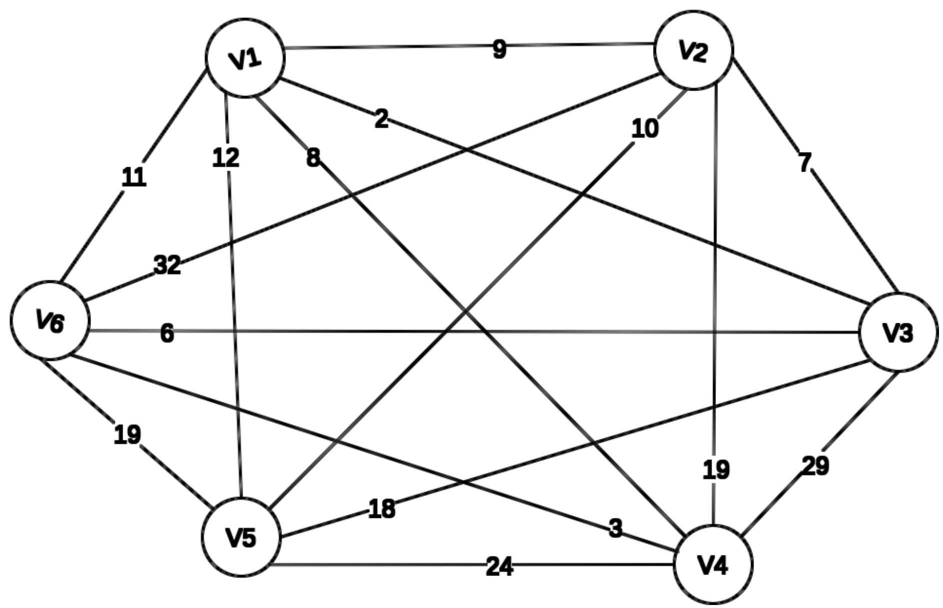
D - E: 10

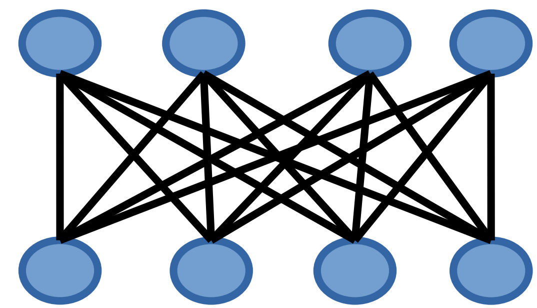
Aplique o Algoritmo do Vizinho mais próximo partindo de ‘A’, o ciclo hamiltoniano obtido é o de menor custo?

Grafos Hamiltonianos

Utilize a heurística do vizinho-mais-próximo para calcular o ciclo hamiltoniano de menor custo nos grafos abaixo:



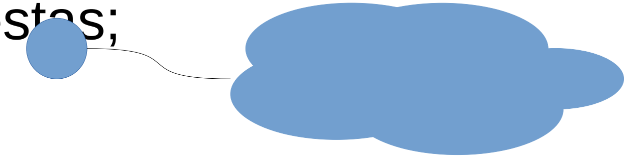




Já foi dito que diferentemente dos Eulerianos (Euler mostrou que um grafo conexo possui ciclo euleriano se e só se nenhum dos seus vértices fossem de grau ímpar) os grafos Hamiltonianos não possuem uma caracterização (uma condição necessária e suficiente: p $\leftrightarrow$ q), porém, existem algumas propriedades que nos ajudam:

(Kenneth Rosen, 2012)

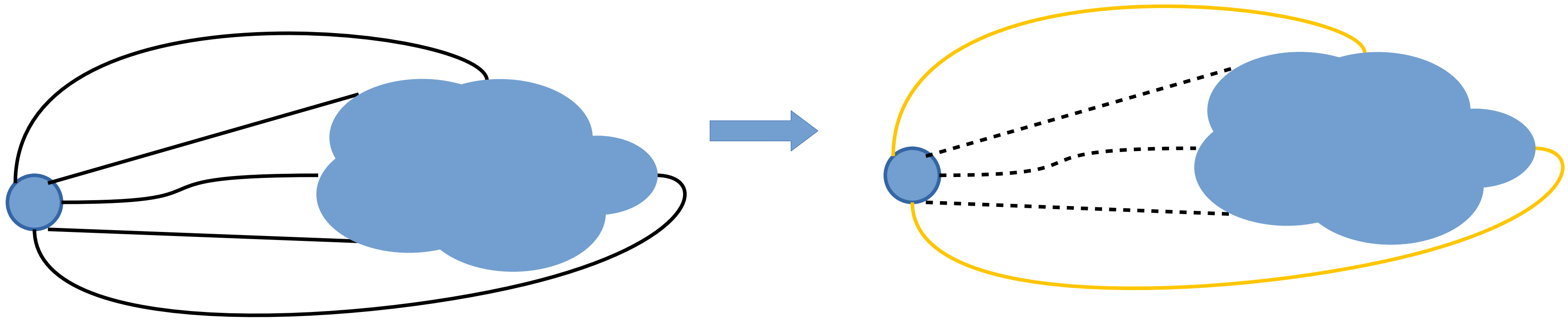
- Um grafo que possua vértice de grau igual a 1 (um) não pode ter um circuito de Hamilton, pois em cada vértice de um circuito de Hamilton incidem duas arestas;



- Se o grafo apresentar vértice de grau dois, então ambas as arestas incidentes neste vértice devem fazer parte de um circuito de Hamilton;

(Kenneth Rosen, 2012)

- Quando um circuito de Hamilton está sendo construído e este circuito passou por um vértice, todas as arestas restantes incidentes a este vértice, exceto as duas usadas no circuito, podem ser desconsideradas;



- O grafo completo K_n possui ciclo hamiltoniano quando $n \geq 3$;
- $K_{n,n}$ é hamiltoniano;

Também existem existem alguns teoremas e condições que nos ajudam:

(Kenneth Rosen, 2012)

Seja G(V,E) um grafo conexo e simples com $n \geq 3$ vértices, seja *deg(v)* o grau de $v \in$ V, duas importantes condições suficientes são:

Teorema de Ore:

Se deg(v) + deg(w) ≥ n para todo par de vértices distintos v e w pertencentes a G e não adjacentes, então G possui um ciclo hamiltoniano.

Teorema de Dirac

Se G é um grafo simples com n vértices n ≥ 3, tal que para todo vértice v em G, deg(v) ≥ n/2, então G possui um ciclo hamiltoniano.

Lembrando que se a condição suficiente:

- For verdadeira (não falhar) → vale a propriedade (exemplo: o grafo possui o ciclo hamiltoniano);
- For falsa (falhar) → nada se pode afirmar sobre a propriedade (o grafo pode ser Hamiltoniano ou não);

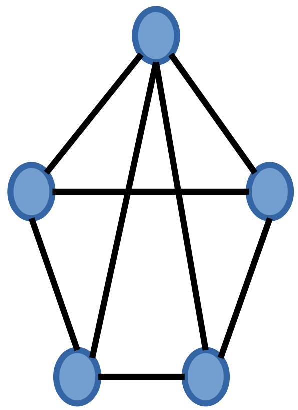
Para cada grafo abaixo, qual atende ao:

Teorema de Dirac;

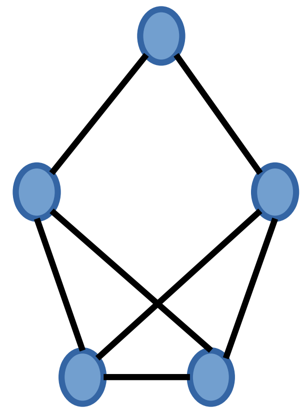
Teorema de Ore

Qual possui ciclo hamiltoniano, enuncie tal ciclo?

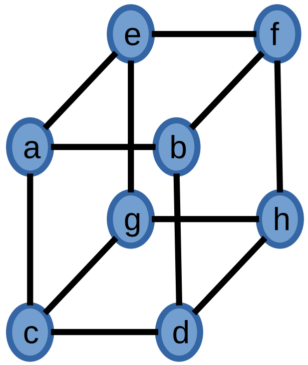
A)



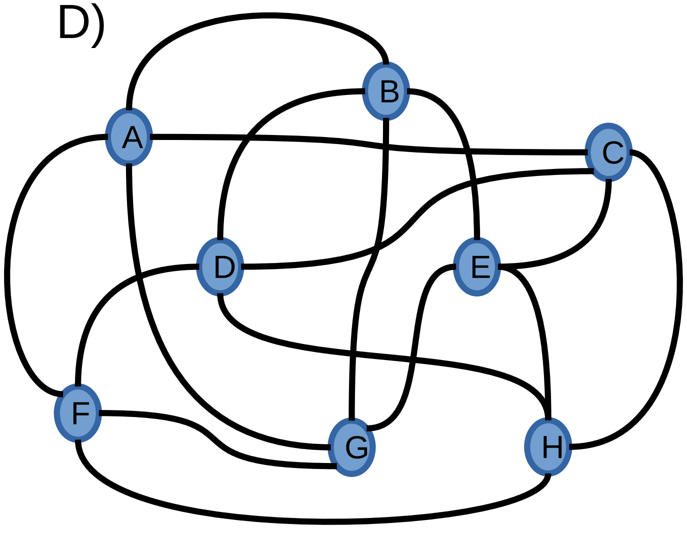
B)



C)



D)



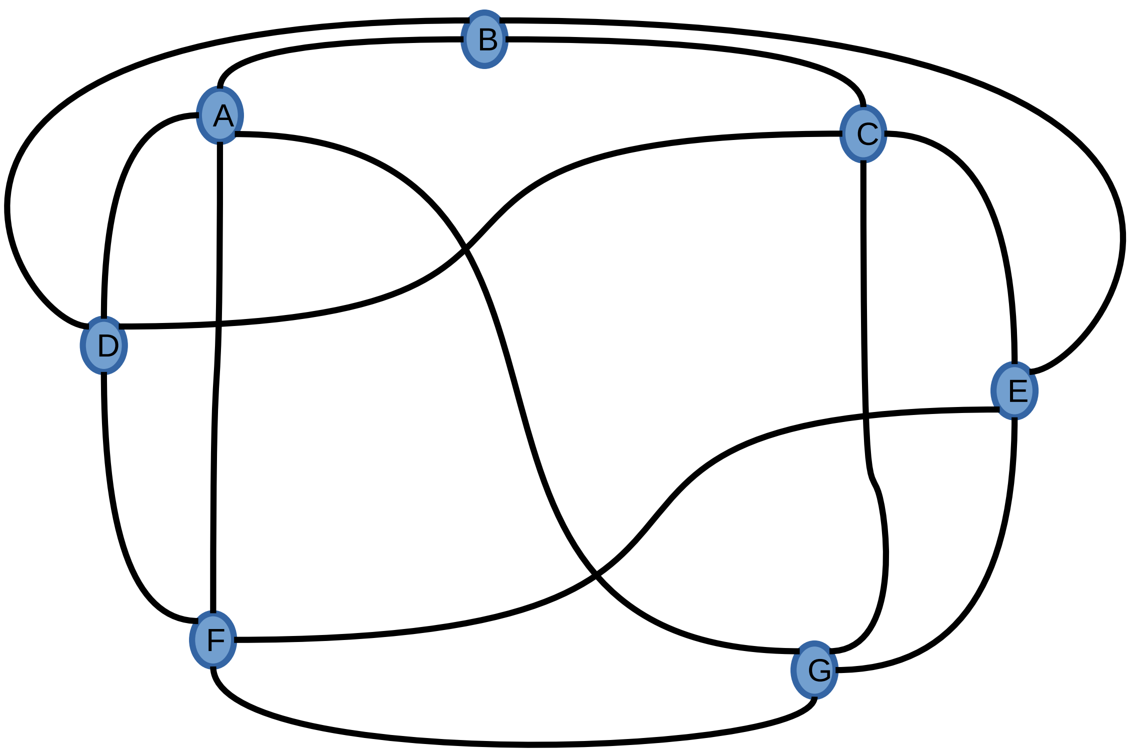
Para cada grafo abaixo, qual atende ao:

Teorema de Dirac;

Teorema de Ore

Ciclo hamiltoniano? Enuncie tal ciclo?

E)



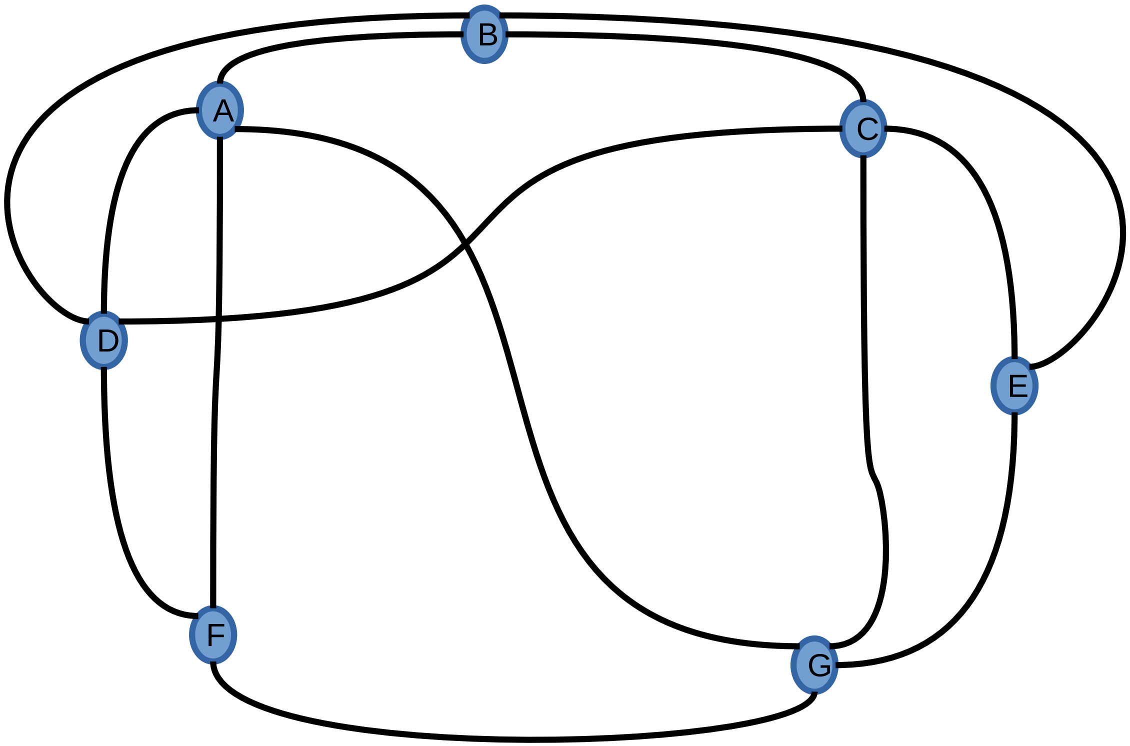
Para cada grafo abaixo, qual atende ao:

Teorema de Dirac;

Teorema de Ore

Ciclo hamiltoniano? Enuncie tal ciclo?

F)

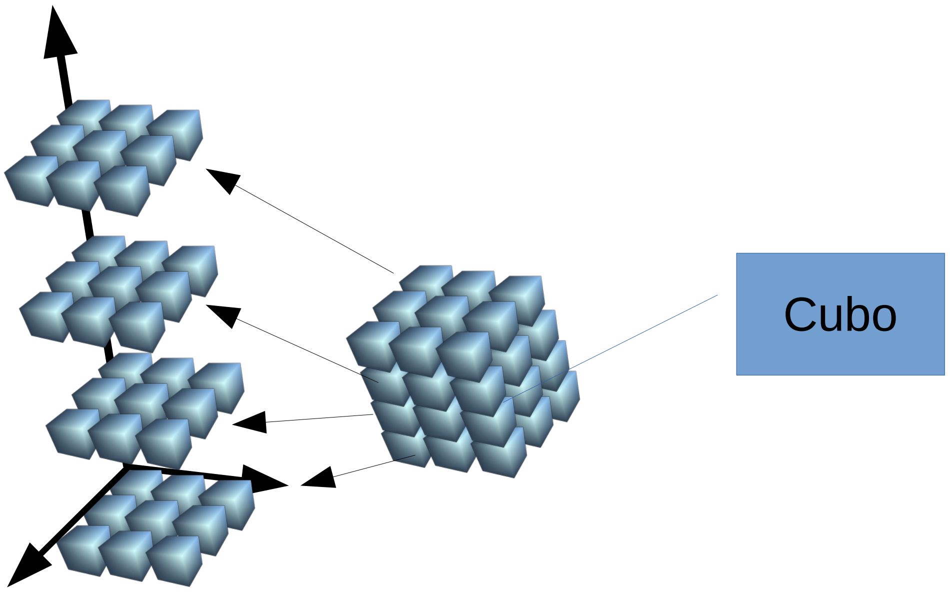


Problema do rato e do queijo:

Um rato come um bloco de 3x3x4 cubos de queijo.

O rato só consegue comer cubos que dividem faces. Ele começa num cubo de um canto, come-o e se move para um cubo adjacente que divide uma face, comendo-o e se movendo para o próximo adjacente, etc. Ignore a gravidade!

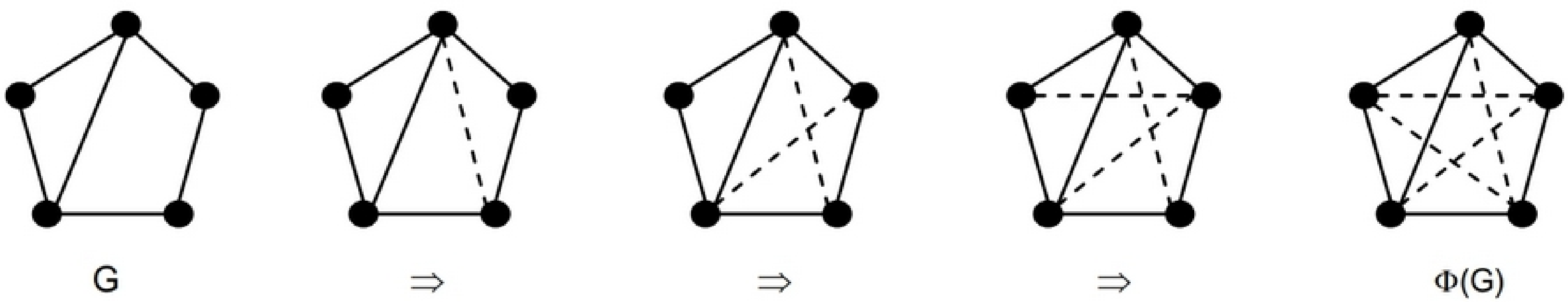
Após comer o último cubo, o rato consegue retornar à posição do primeiro cubo comido. Este ciclo é hamiltoniano?



(Netto,Boaventura & Jurkiewicz, Samuel, 2017)

Definição: O fecho hamiltoniano φ(G) de um grafo G é o grafo obtido ao final de uma sequência de adições sucessivas de arestas a todo par v, w | d(v) + d(w) >= n, onde v e w são não adjacentes e d(v) e d(w) são graus vigentes, na sequência de grafos assim obtida.

Observe nos grafos abaixo a construção progressiva do fecho:

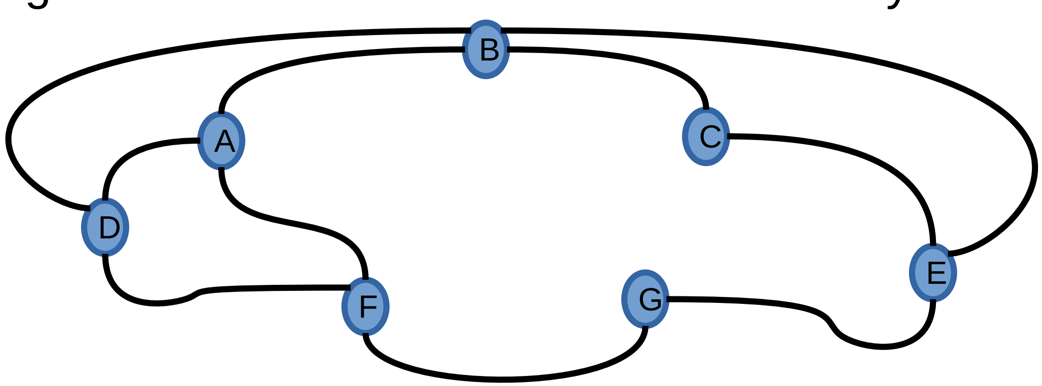


Teorema 8.8 (Bondy e Chvátal): Se φ(G) = Kn, então G será hamilltoniano.

Uma versão modificada deste teorema afirma (Teorema 8.9) que um grafo G será hamilltoniano se e somente se φ(G) for hamiltoniano.

Verifique que o grafo abaixo satisfaz o teorema de Bondy-Chvátal, mas não os de Dirac e Ore.

H)



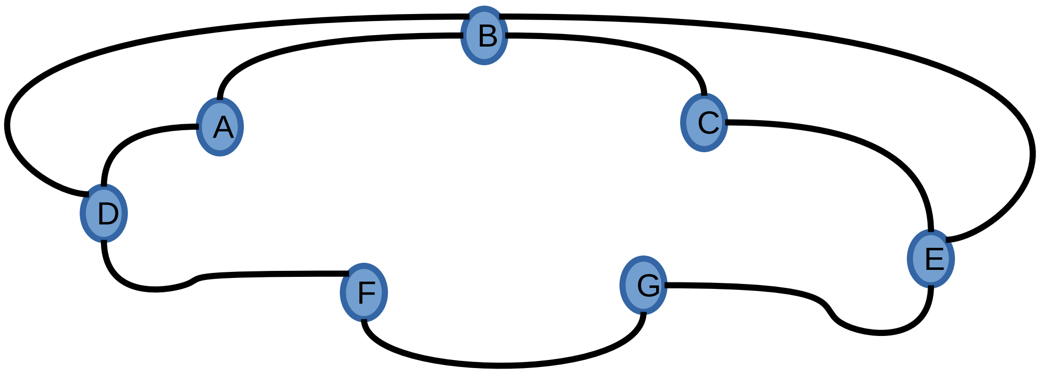
(Netto,Boaventura & Jurkiewicz, Samuel, 2017)

Observe, ainda, que o Teorema 8.9 economiza trabalho: você pode parar o processo de construção do fecho assim que o grafo obtido satisfaça, por exemplo, o teorema de Ore.

O Teorema 8.9 não é condição necessária e suficiente. Apenas, ele relaciona o grafo com o seu fecho, o que nos deixa o problema de saber se o fecho é, ou não, hamiltoniano.

Aplique o procedimento de parar o processo de construção do fecho assim que o grafo obtido satisfaça, por exemplo, o teorema de Ore (ou Dirac ou Bondy e Chvátal)

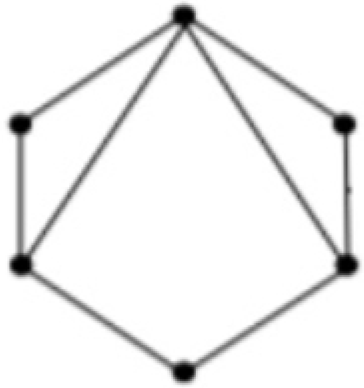
I)



(Netto,Boaventura & Jurkiewicz, Samuel, 2017)

Para o grafo abaixo, responda justificando:
a) Ele é euleriano? Porquê?
b) Ele é hamiltoniano? Porquê?
c) Mostre que via os teoremas de Ore e Dirac não podemos provar que o grafo é hamiltoniano.
d) Mostre que via o teorema de Bondy-Chvátal é possível mostrar que o grafo é hamiltoniano.

J)



TEG

Bibliografia

Básica

LUCCHESI, C. L. et alli. Aspectos Teóricos da Computação, Parte C: Teoria dos Grafos, projeto Euclides, 1979.

SANTOS, J. P. O. et alli. Introdução à Análise Combinatória. UNICAMP; 1995.

SZWARCFITER, J. L. Grafos e Algoritmos Computacionais. Campus, 1986.

GERSTING, Judith L. Fundamentos Matemáticos para a Ciência da Computação. Rio de Janeiro. 3a Ed. Editora.

Complementar:

1.) CORMEN, T. Introduction to Algorithms, third edition, MIT press, 2009

2.) ROSEN, K. Discrete Mathematics and its applications, seventh edition, McGraw Hill, 2011.

3.) WEST, Douglas, B. Introduction to Graph Theory, second edition, Pearson, 2001.

4.) BONDY, J.A., MURTY, U.S.R., Graph Theory with applications , Springer, 1984.

5.) SEDGEWICK, R. Algorithms in C - part 5 - Graph Algorithms, third edition, 2002, Addison-Wesley.

6.) GOLDBARG, M., GOLDBARG E., Grafos: Conceitos, algoritmos e aplicações. Editora Elsevier, 2012.

7.) BONDY, J.A., MURTY, U.S.R., Graph Theory with applications , Springer, 1984

8.) FEOFILOFF, P., KOHAYAKAWA, Y., WAKABAYASHI, Y., uma introdução sucinta à teoria dos grafos. 2011. (www.ime.usp.br/~pf/teoriadosgrafos)

9.) DIESTEL, R. Graph Theory, second edition, springer, 2000

10.) FURTADO, A. L. Teoria de grafos. Rio de janeiro. Editora LTC. 1973.

11.) WILSON, R.J. Introduction to Graph Theory. John Wiley & Sons Inc., 1985

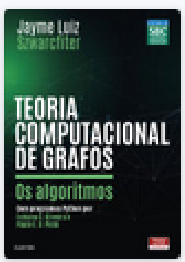
12.) BOAVENTURA NETTO , P. O. Grafos: Teoria, Modelos, Algoritmos. Edgard Blucher, SP, quinta edição

Tutoriais, artigos, notas de aula...

TEG

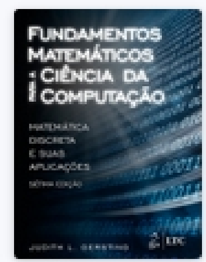
Vários livros podem ser acessados np formato eletrônico (e-book) via https://www.udesc.br/bu/acervos/ebook

NETTO, Paulo Oswaldo B.; JURKIEWICZ, Samuel. Grafos: introdução e prática. [Digite o Local da Editora]: Editora Blucher, 2017. E-book.



Teoria Computacional de Grafos - Os Algoritmos

Jayme Luiz Szwarcfiter

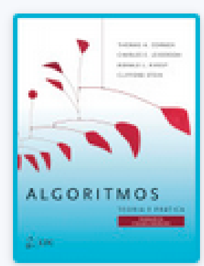


Fundamentos Matemáticos para a Ciência da Computação

Judith L. Gersting

Grafos

Marco Goldbarg



Algoritmos - Teoria e Prática

Thomas Cormen