Matemática das Coisas

Parte 4

Grafos

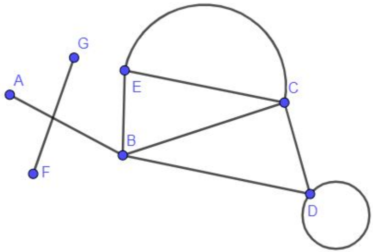
Aula de 21 de Novembro de 2023

José Joaquim Martins Oliveira

Grafos – Definição

Um grafo $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ consiste em

- conjunto $\mathcal{V}$, finito e não vazio, de pontos chamados vértices
- conjunto $\mathcal{A}$, possivelmente vazio, de ligações entre pares de vértices, chamadas arestas.



Vértices

$$\mathcal{V} = \{A, B, C, D, E, F, G\}$$

Arestas

$$\begin{aligned} \mathcal{A} \;=\; & \{(AB), (BE), (EC), \\ & (EC), (CB), (CD), \\ & (DD), (DB), (FG)\} \end{aligned}$$

Os grafos têm inúmeras aplicações,

sendo estrutura ideais para representar

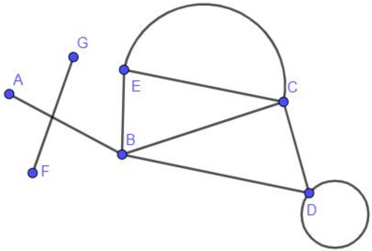
- redes de transporte
- relações sociais
- canais de comunicação
- labirintos
- conexões em geral

As arestas podem representar

- ruas, canalizações, linha elétricas, vias de comunicação
- relações de amizade, ligações entre servidores,
- e muitos outras "ligações" ...

Grafos – Noções gerais

No grafo $\mathcal{G}=(\mathcal{V},\mathcal{A})$ onde $\mathcal{V}=\{A,B,C,D,E,F,G\}$ e $\mathcal{A}=\{(AB),(BE),(EC),(EC),(CB),(CD),(DD),(DB),(FG)\}$

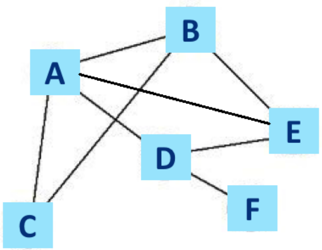


- A aresta (DD) chama-se lacete;
- As arestas (EC) dizem-se paralelas;
- Vértices ligados por arestas dizem-se adjacentes;
- Arestas que ligam um vértice, V, dizem-se incidentes em V.

Grafos – Noções gerais

Dado um grafo $\mathcal{G} = (\mathcal{V}, \mathcal{A})$

- chamamos grau de um vértice V, e representa-se por $g(V)$, ao número de arestas incidentes no vértice V



Graus

$g(A) = 4$ $\quad$ $g(D) = 3$

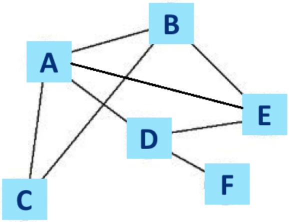
$g(B) = 3$ $\quad$ $g(E) = 3$

$g(C) = 2$ $\quad$ $g(F) = 1$

O grau pode ter vários significados, consoante o contexto.

Num grafo, definimos

• passeio como uma sequência de vértices, possivelmente não todos distintos, tais que cada dois vértices consecutivos define uma aresta.
• trajecto (ou trilho) como um passeio onde não há repetição de arestas.
• caminho como um passeio que não repete vértices (nem arestas)
• circuito como um passeio que começa e acaba no mesmo vértice.
• circuito simples como um circuito que mas não repete arestas.
• ciclo como um circuito simples, onde não há repetição de vértices com a excepção dos vértices inicial e final.



ABEDA é um ciclo

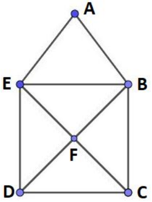
ACBADE é um trajecto, mas não é caminho

ABEDABCA é um circuito não simples (também não é ciclo)

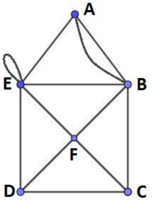
EDABCAE é um circuito simples, mas não é ciclo

Tipos de Grafos

- Grafo simples quando não possui lacetes nem arestas paralelas

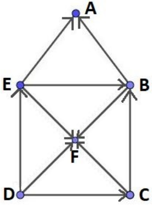


Grafo simples

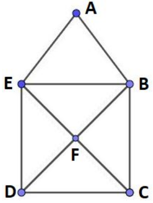


Grafo não simples

- Grafo orientado quando todas as arestas têm uma orientação. As setas podem representar sentido no trânsito, quando as arestas representam estradas



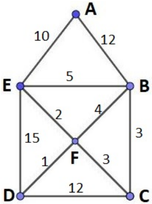
Grafo orientado



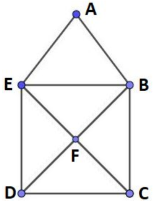
Grafo não orientado

Tipos de Grafos

• Grafo ponderado quando todas as arestas têm um **peso**, isto é um número que poderá representar uma distância, um custo, um tempo, etc..

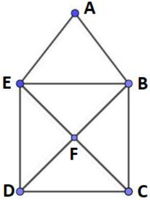


Grafo ponderado

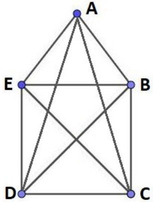


Grafo não ponderado

• Grafo planar quando é possível representa-lo no plano sem que as arestas se cruzem



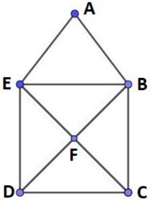
Grafo planar



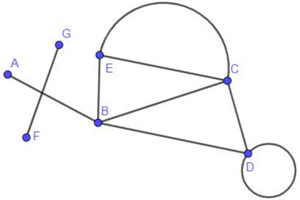
Grafo não planar

Tipos de Grafos

- Grafo conexo se qualquer par de vértices está ligado por um passeio
  No caso contrário, o grafo é desconexo



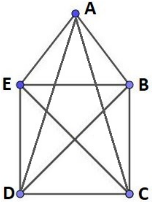
Grafo conexo



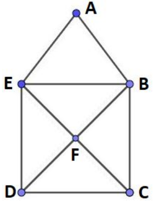
Grafo desconexo

2 componentes conexas

- Grafo completo se todo o par de vértices define uma aresta

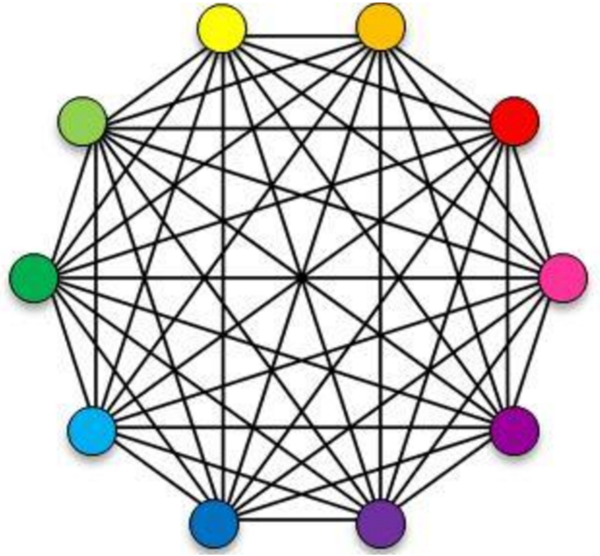


Grafo completo (K_5)



Grafo não completo

Grafo completo com 10 vértices – K_{10}

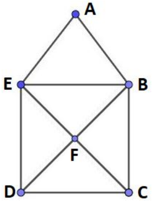


Grafos de Euler (eulerianos)

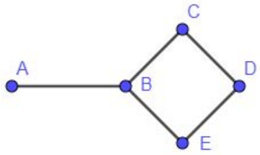
- Um trajecto diz-se euleriano se percorre todas as arestas.

Quais dos seguintes grafos admite um trajecto euleriano?

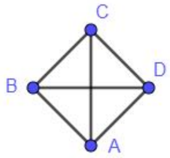
G_1



G_2



G_3



G_1 e G_2

- **Quando é que um grafo admite um trajecto euleriano?**

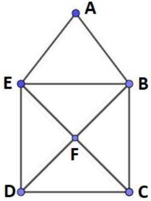
Quando o grafo é conexo e possui, no máximo, dois vértices de grau ímpar.

Grafos de Euler (eulerianos)

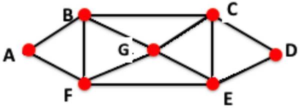
• Um grafo diz-se de Euler ou euleriano se possui um circuito simples que percorre todas as arestas. No caso contrário, o grafo diz-se não euleriano

Quais dos seguintes grafos é euleriano?

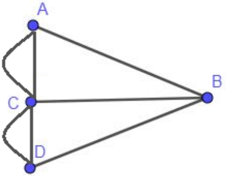
$\mathbf{G_1}$



$\mathbf{G_2}$



$\mathbf{G_3}$



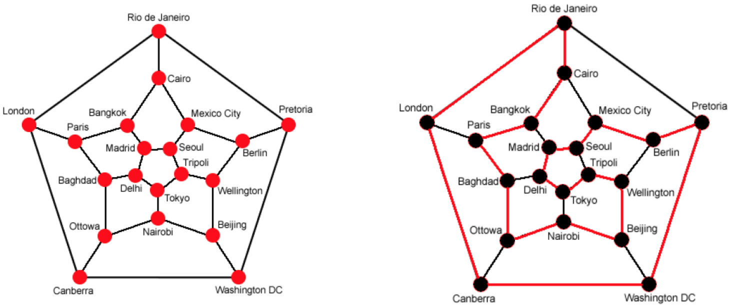
Apenas $\mathbf{G_2}$

Quando é que um grafo é euleriano?

Quando o grafo é conexo e todos os vértices têm grau par.

Grafo de Hamilton (hamiltoniano)

- Um grafo diz-se de Hamilton ou hamiltoniano se possui um ciclo que contém todos os vértices. No caso contrário, o grafo diz-se não hamiltoniano

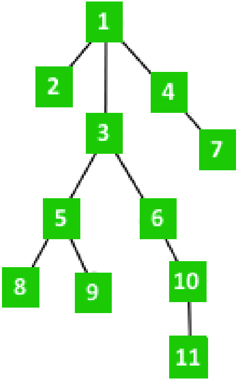


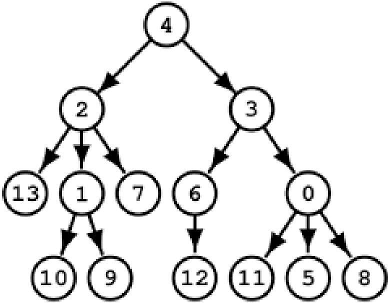
Este grafo será hamiltoniano? **SIM**

Não existe uma caracterização completa de grafos hamiltonianos

Árvore

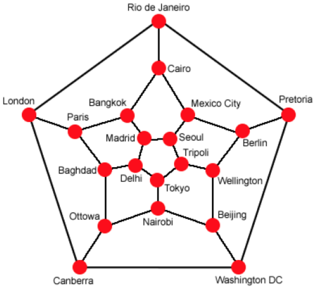
- Árvore é um grafo conexo sem ciclos





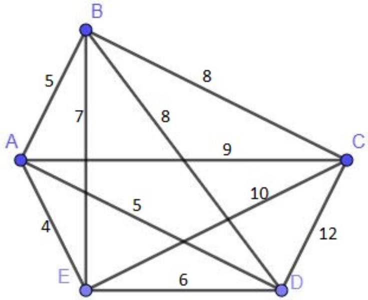
Árvore abrangente

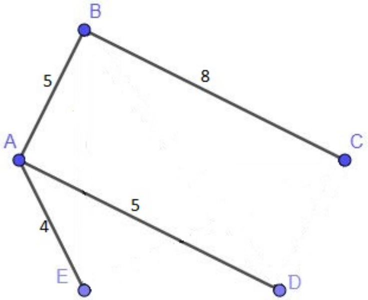
- Uma árvore abrangente de um grafo é uma árvore que contém todos os vértices desse grafo.



Árvore abrangente mínima

• Uma árvore abrangente mínima de um grafo ponderado é uma árvore abrangente desse grafo em que a soma dos pesos das suas arestas é mínima.





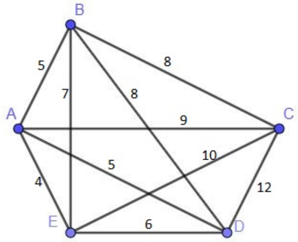
Árvore abrangente mínima

- A obtenção de uma árvore abrangente mínima soluciona diversos problemas de optimização, tais como:
  - Implementação de redes de distribuição de energia numa certa localidade/cidade/país;
  - Construção de redes de saneamento básico;
  - Construção de redes de abastecimento de água;
  - etc...
- **Como obter uma árvore abrangente mínima?**
  - Algoritmo de Kruskal;
  - Algoritmo de Prim.

Algoritmo de Kruskal

1. Ordenar as arestas por ordem crescente de pesos;
2. Respeitando a ordem estabelecida no ponto 1, acrescentar cada aresta à árvore se esta não criar um ciclo;
3. Terminar quando se tiver obtido uma árvore abrangente.

Algoritmo de Kruskal (exemplo)



- Primeiro ordenar as arestas por ordem crescente de pesos:

  $(4)AE; (5)AB; (5)AD; (6)DE; (7)EB; (8)BD; (8)BC; (9)AC; (10)CE; (12)DC$

Algoritmo de Kruskal (exemplo)

- Primeiro ordenar as arestas por ordem crescente de pesos:

  $(4)AE; (5)AB; (5)AD; (6)DE; (7)EB; (8)BD; (8)BC; (9)AC; (10)CE; (12)DC$

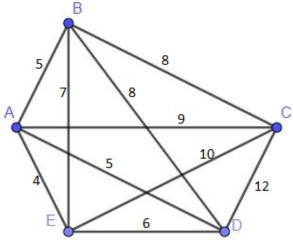
- Segundo usar as arestas

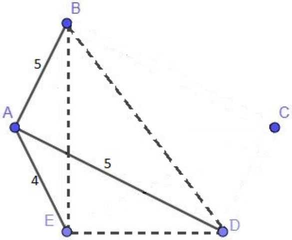
$$(4)AE; (5)AB; (5)AD$$

e rejeitar as arestas

$$(6)DE; (7)EB; (8)BD$$

por formarem ciclos





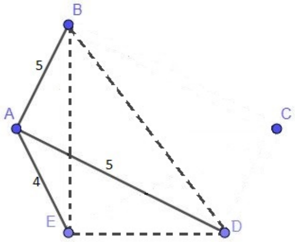
Algoritmo de Kruskal (exemplo)

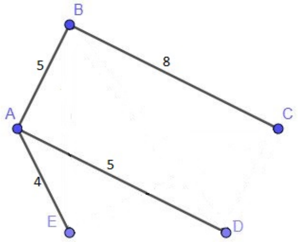
- Primeiro ordenar as arestas por ordem crescente de pesos:

  $(4)AE; (5)AB; (5)AD; (6)DE; (7)EB; (8)BD; (8)BC; (9)AC; (10)CE; (12)DC$

- Segundo usar as arestas $(4)AE; (5)AB; (5)AD$ e rejeitar as arestas $(6)DE; (7)EB; (8)BD$ por formarem ciclos.

- Terceiro usar a aresta $(8)BC$.



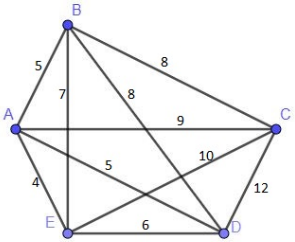


- O processo terminou pois já temos todos os vértices em conexão.

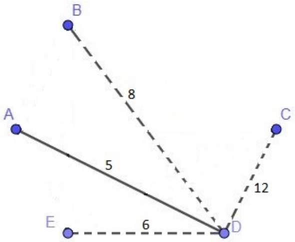
Algoritmo de Prim

1. Escolher um vértice qualquer;
2. De entre todas as arestas que incidem nesse vértice, escolher a de menor peso;
3. De entre todas as arestas que ligam algum dos vértices que já pertence à árvore a um que não pertence, escolher a de menor peso;
4. Repetir o item anterior até se obter uma árvore abrangente.

Algoritmo de Prim (exemplo)

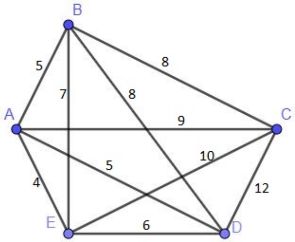


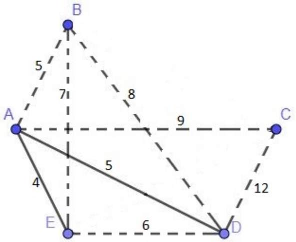
- Escolha-se um vértice qualquer, por exemplo D.
- Partindo de D, escolhe-se a aresta de menor peso. No caso AD.



Algoritmo de Prim (exemplo)

- Escolha-se um vértice qualquer, por exemplo D.
- Partindo de D, escolhe-se a aresta de menor peso. No caso AD.
- Analisando as arestas que incidem em A e em D, escolhe-se a que tem menor peso.

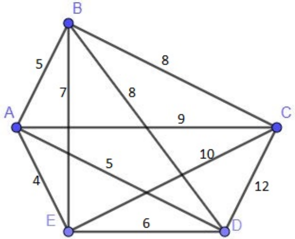


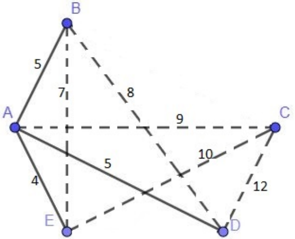


No caso AE

Algoritmo de Prim (exemplo)

- Escolha-se um vértice qualquer, por exemplo D.
- Partindo de D, escolhe-se a aresta de menor peso. No caso AD.
- Analisando as arestas que incidem em A e em D, escolhe-se a que tem menor peso. No caso AE.
- Analisando as arestas que incidem em A, em D e em E, escolhe-se a que tem menor peso.

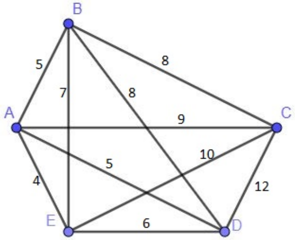


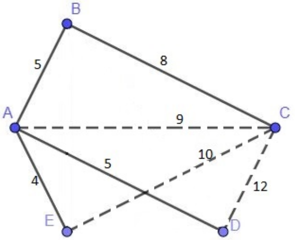


No caso AB.

Algoritmo de Prim (exemplo)

- Escolha-se um vértice qualquer, por exemplo D.
- Partindo de D, escolhe-se a aresta de menor peso. No caso AD.
- Analisando as arestas que incidem em A e em D, escolhe-se a que tem menor peso. No caso AE.
- Analisando as arestas que incidem em A, em D e em E, escolhe-se a que tem menor peso. No caso AB.
- Faltando o vértice C, escolhe-se a aresta com menor peso entre as que incidem em C.

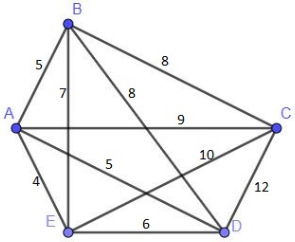


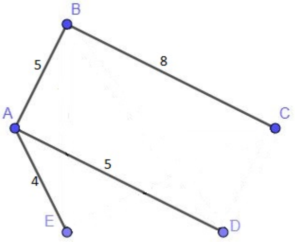


No caso BC.

Algoritmo de Prim (exemplo)

- Escolha-se um vértice qualquer, por exemplo D.
- Partindo de D, escolhe-se a aresta de menor peso. No caso AD.
- Analisando as arestas que incidem em A e em D, escolhe-se a que tem menor peso. No caso AE.
- Analisando as arestas que incidem em A, em D e em E, escolhe-se a que tem menor peso. No caso AB.
- Faltando o vértice C, escolhe-se a aresta com menor peso entre as que incidem em C. No caso BC.
- Obtém-se





Problema do Caixeiro viajante

- Problemas de optimização, tais como:
  - Estabelecer percursos para a distribuição do correio;
  - Definir percursos de distribuição de mercadorias por uma cadeia de supermercados;
  - Planear uma viagem cujo objectivo é visitar várias cidades e regressar ao ponto de partida;
  - etc...

são conhecidos como problema do caixeiro viajante.

- Concretamente: Um viajante tem de visitar um certo número de cidades e voltar à cidade de partida.

Qual será a forma mais económica de o fazer?

Problema do Caixeiro viajante

• A solução ideal será obter um ciclo hamiltoniano (ciclo que contém todos os vértices de um grafo) com custo mínimo.

- Para o obter será necessário analisar todos os casos possíveis.
- Sendo um método exaustivo, pode tornar-se difícil, ou mesmo impossível, de lidar se o número de locais a visitar aumentar significativamente.
- Existem algoritmos específicos (não exaustivos) para tratar o problema do caixeiro viajante, mas não há a garantia de obter a solução óptima:
  - algoritmo dos mínimos sucessivos.
  - algoritmo por ordenação dos pesos das arestas.

Problema do Caixeiro viajante

Algoritmo dos mínimos sucessivos

• Começa-se por um vértice e escolhe-se sempre a aresta com menor peso até construir um ciclo hamiltoniano.
• Realiza-se a operação anterior começando em todos os vértices.
• A solução encontrada corresponde à que tem a menor soma dos pesos das arestas que compõem o ciclo.

Atenção: Encontra-se uma boa solução, mas esta pode não corresponder à solução óptima do problema.

Problema do Caixeiro viajante

Algoritmo por ordenação dos pesos das arestas

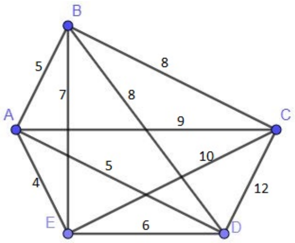
• Começa-se por ordenar as arestas por ordem crescente dos pesos.

• Escolhe-se sucessivamente a aresta com menor peso, respeitando as seguinte regras:

- Não se pode escolher três arestas que incidem no mesmo vértice;
- Não se pode fechar o circuito quando ainda restarem vértice não visitados.

Atenção: Analogamente ao algoritmo anterior, provavelmente encontra-se uma boa solução, mas esta pode não corresponder à solução óptima do problema.

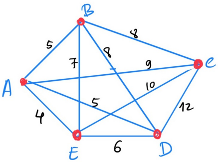
Problema do Caixeiro viajante (exemplo)

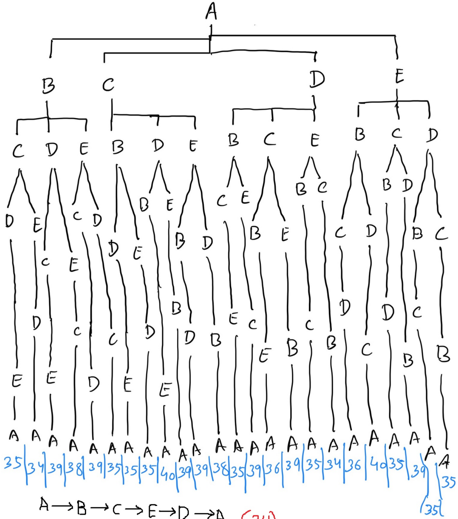


- Sendo o grafo um K_5, então o método exaustivo para solucionar o problema do caixeiro viajante envolve a análise de

$$(5-1)! = 4! = 4 \times 3 \times 2 \times 1 = 24$$

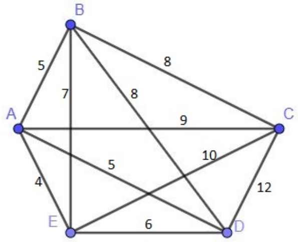
casos.
- O estudo exaustivo de um grafo completo com n vértices ($n \in \mathbb{N}$), isto é um K_n, exige a análise de para $(n-1)!$ casos para solucionar o problema do caixeiro viajante.





$A \rightarrow B \rightarrow C \rightarrow E \rightarrow D \rightarrow A$ (34)

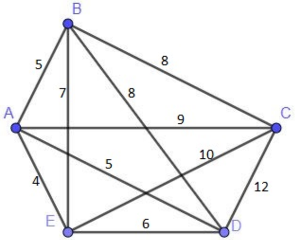
Algoritmo dos mínimos sucessivos (exemplo)



$$
\begin{array}{ccccccccccccl}
A & \xrightarrow{4} & E & \xrightarrow{6} & D & \xrightarrow{8} & B & \xrightarrow{8} & C & \xrightarrow{9} & A & \color{red}{\text{custo } 35} \\
B & \xrightarrow{5} & A & \xrightarrow{4} & E & \xrightarrow{6} & D & \xrightarrow{12} & C & \xrightarrow{8} & B & \color{red}{\text{custo } 35} \\
C & \xrightarrow{8} & B & \xrightarrow{5} & A & \xrightarrow{4} & E & \xrightarrow{6} & D & \xrightarrow{12} & C & \color{red}{\text{custo } 35} \\
D & \xrightarrow{6} & E & \xrightarrow{4} & A & \xrightarrow{5} & B & \xrightarrow{8} & C & \xrightarrow{12} & D & \color{red}{\text{custo } 35} \\
E & \xrightarrow{4} & A & \xrightarrow{5} & D & \xrightarrow{8} & B & \xrightarrow{8} & C & \xrightarrow{10} & E & \color{red}{\text{custo } 36}
\end{array}
$$

- a solução dada pelo algoritmo dos mínimos sucessivos pode ser qualquer dos 4 primeiros ciclos obtidos, pois todos têm um custo de 35.

Algoritmo por ordenação do peso das arestas (exemplo)



- Começa-se por ordenar as arestas por ordem crescente dos pesos:

$$(4)AE; (5)AB; (5)AD; (6)DE; (7)EB; (8)BD; (8)BC; (9)AC; (10)CE; (12)DC$$

Seguindo o algoritmo tem-se

$(4)AE; (5)AB; \cancel{(5)AD}; (6)DE; \cancel{(7)EB; (8)BD}; (8)BC; \cancel{(9)AC; (10)CE}; (12)DC$

- A solução dada pelo algoritmo por ordenação do peso das arestas é

$$AEDCBA$$

cujo custo é 35.