

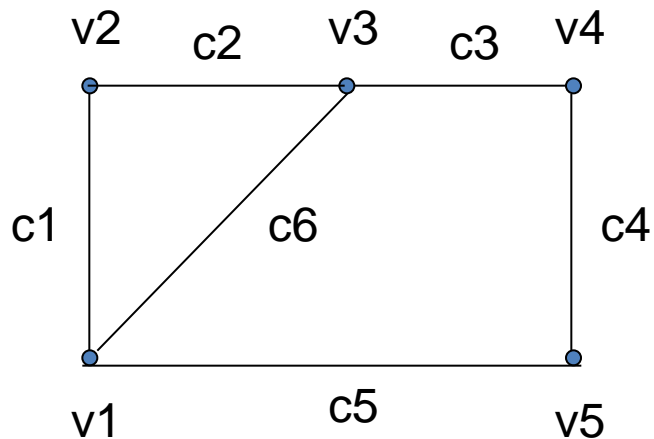
Grafos

IFRN

Prof. Robinson Alves

Caminhos

- É uma seqüência de arestas onde o vértice final de uma aresta é o vértice inicial da próxima



$\{c1, c2, c4, c5, c6\}$ $\{c2, c3, c4, c5\}$

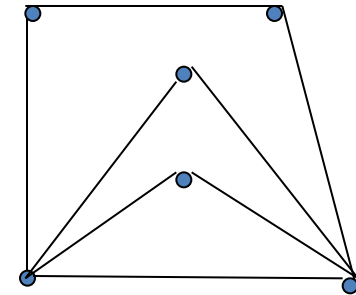
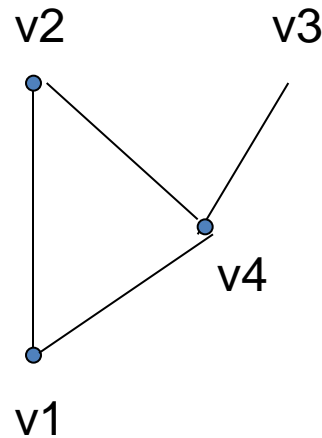
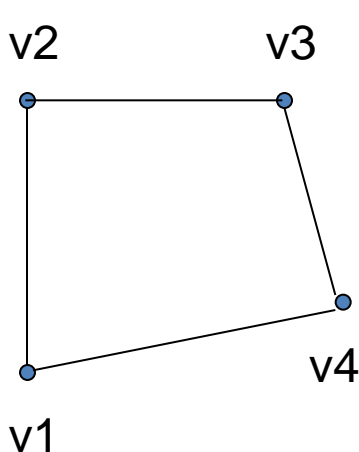
$\{v1, v2, v3, v4, v5\}$ $\{v2, v3, v1, v5, v4\}$

Caminhos

- Um caminho de k vértices tem $(k-1)$ arestas, $v_1v_2, v_2v_3, \dots, v_{k-1}v_k$ e o comprimento do caminho é $(k-1)$
 - Caminho aberto (simples ou elementar): Todos os vértices são distintos. Ex.: $\{c_2, c_3, c_4\}$
 - Caminho fechado ou ciclo: é aquele em que o vértice inicial é igual ao final, ou seja: $v_1, v_2, \dots, v_k, v_{k+1}$, onde $v_{k+1} = v_1$. Ex $\{c_1, c_2, c_3, c_4, c_5\}$ não é um caminho aberto

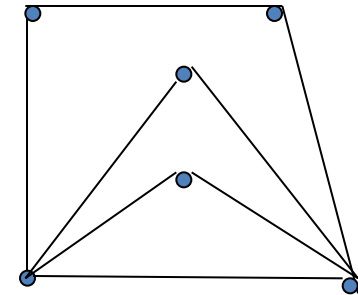
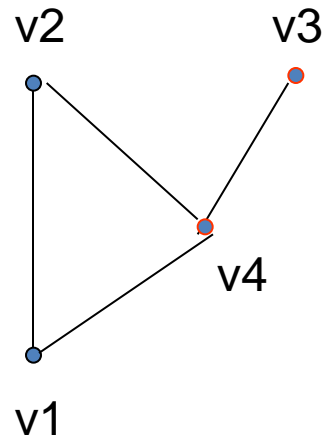
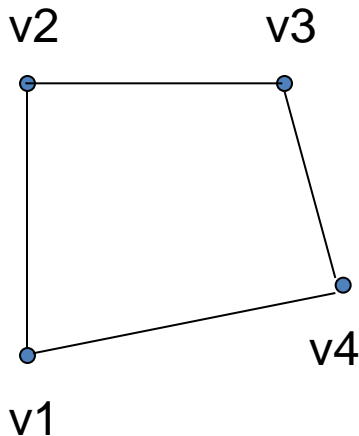
Caminhos

- Grafos Eulerianos (grafos de Euler)
 - São grafos onde é possível achar um caminho fechado (ciclo), passando em cada aresta uma única vez
 - Quais são os grafos de Euler?



Caminhos

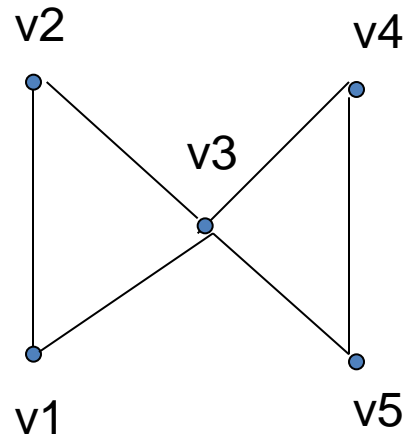
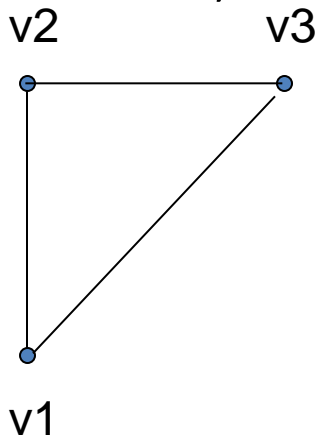
- Grafos Eulerianos (grafos de Euler)
 - Teorema: um grafo conexo G é um grafo de Euler se e somente se todos os seus vértices são grau par.



Caminhos

- Caminhos Eulerianos

- São caminhos que passam por cada aresta um vez, passando por todas
- Teorema: baseado no grau dos vértices do grafo: existe um caminho Euleriano em um grafo se:
 - Não ocorrer vértices de grau ímpar (inicia e termina no mesmo vértice)

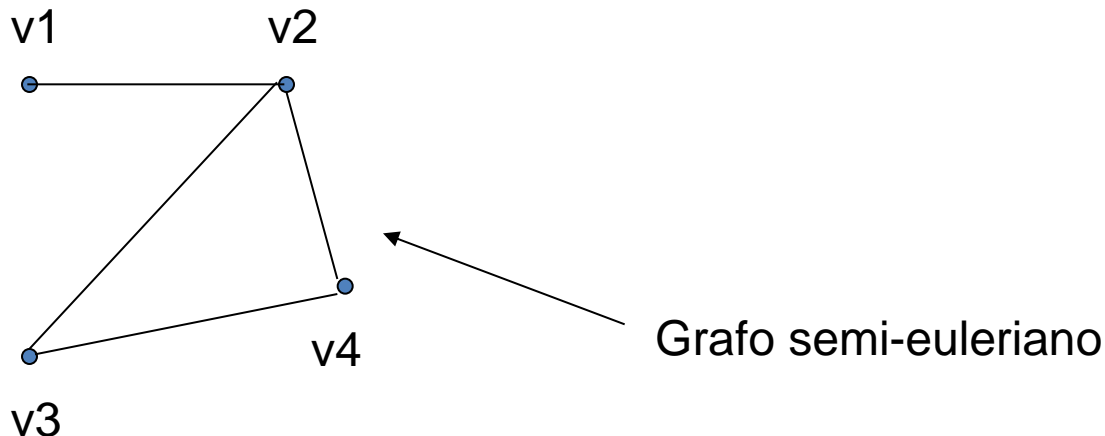


Caminhos

- Caminhos Eulerianos

- Teorema: baseado no grau dos vértices do grafo: existe um caminho Euleriano em um grafo se:

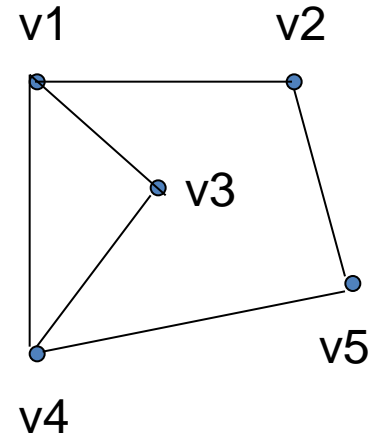
- Existem 2 vértices de grau ímpar (inicia em um vértice ímpar e termina em outro vértice ímpar)



Caminhos

- Algoritmo para verificar a existência de um caminho euleriano

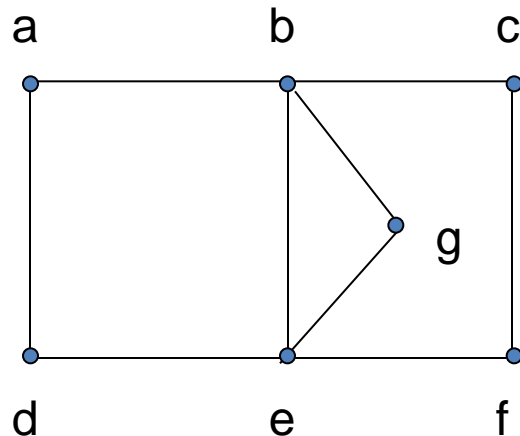
```
grau=0; soma=0; matadj[][]; N=numero de linhas da matriz; f=0;//linha atual;
Enquanto (soma<=2) e (f<=N) {
    grau=0;
    para (g=0;g<N;g++) {
        grau+=matadj[f][g];
    }
    se grau mod 2 == 1 // ímpar
        soma++
    f++;
}
Se (soma>2)          NÃO EXISTE CAMINHO
Senao                EXISTE CAMINHO
```



	1	2	3	4	5	<i>grau</i>
1	0	1	0	1	0	2
2	1	0	0	1	1	3
3	1	0	0	1	0	2
4	0	1	1	0	1	3
5	0	1	0	1	0	2

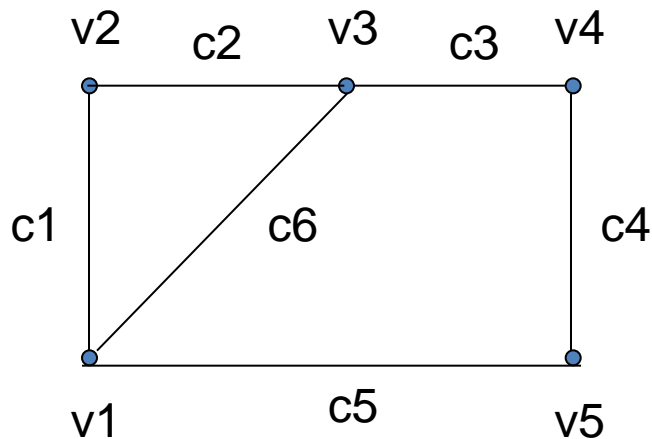
Caminhos

- Método de Fleury para traçar um ciclo euleriano
 - Inicie em qualquer vértice v e atravesse as arestas de uma maneira arbitrária, seguindo as seguintes regras
 - R1) apague a aresta que foi visitada e se algum vértice ficar isolado, apague-o também.
 - R2) em cada estágio, nunca atravesse uma aresta, se naquele momento a sua remoção divide o grafo em duas ou mais componentes não triviais (excluindo os vértices isolados).



Caminhos

- Ciclos e Caminhos Hamiltonianos
 - Um ciclo hamiltoniano em um grafo conexo G é definido com um caminho simples fechado em que cada vértice de G é visitado uma única vez (com exceção do nó inicial == final)
 - Um ciclo hamiltoniano em um grafo de n vértices tem n arestas



- Não é um grafo de euler
- Possui ciclo hamiltoniano?

Caminhos

- Condições para existência de um caminho Euleriano em um grafo
 - Grafo de euler ou semi-euler
- Condições para existência de um caminho Hamiltoniano em um grafo
 - Não tem solução

Problema do Carteiro Chinês

- A primeira referência ao problema foi em 1962 em uma revista chinesa, tendo a designação ficada associada ao problema
- Pode ser formulado como o problema de encontrar um ciclo Euleriano de menor custo
 - Exemplo: Considere um bairro de uma cidade em que um carteiro é responsável por distribuir correspondências. Neste modelo as arestas ponderadas representam as ruas e suas respectivas distâncias e os vértices os cruzamentos. Para que a correspondência seja entregue, é necessário que todas as ruas sejam percorridas pelo menos uma vez e que ao final o carteiro volte a estação do correio. O problema consiste em determinar o menor caminho possível para que o carteiro faça seu trabalho.
 - Problemas similares: recolhimento do lixo, inspeção de linhas de transmissão, inspeção de linhas telefônicas, etc.

Problema do Carteiro Chinês

- Encontrar o menor caminho (sendo de euler)
 - Problema do carteiro chinês: início ao fim passando em todas as arestas uma vez (menor caminho) – caminho de euler
 - Determinar o custo do menor caminho
 - Solução 1: se o grafo for euleriano, aplicar algoritmo de Fleury
 - Solução 2: se o grafo não for de euler aplicar o algoritmo de carteiro chinês (Christofides)

Algoritmo do carteiro chinês

P1: determinar os vértices de grau ímpar

P2: construa a matriz de distancia D , com os vértices de grau ímpar (Dijkstra)

P3: determinar o par de vértices c / menor caminho através da matriz D

P4: construa um caminho artificial de v_i para v_j com custo encontrado em P3

P5: elimine de D as colunas e linhas correspondentes v_i e v_j

P6: se houver linhas e colunas em D , volte para P3

P7: ache o caminho de euler (fleury)

P8: o custo será igual à soma dos custos de todas as arestas acrescida dos custos das arestas encontradas em P3

Problema do Menor Caminho

- Algoritmo Dijkstra
 - Edsger Dijkstra, ganhador do prêmio Turing em 1972
 - A idéia principal é o uso de um método guloso para o problema do caminho mínimo com origem única.
 - Funciona em grafo dirigido ou não dirigido com arestas de peso não negativo, em tempo computacional $O((m+n)\log n)$ onde m é o número de arestas e n é o número de vértices

Problema do Menor Caminho

- Algoritmo Dijkstra -determinar:
 - Seja $G(V,A)$ e uma função $L(V,W)$ pertencente aos reais e um vértice fixo v_0 de V (fonte)
 - Problema: determinar v_0, v_1, \dots, v_k , tal que $\sum_{i=0}^{k-1} l(v_i, v_{i+1})$ seja mínimo

$$L(v_i, v_j) = \begin{cases} \infty & \text{se não existir aresta}(v_i, v_j) \\ 0 & v_i = v_j \\ \text{custo} & \text{se existir aresta}(v_i, v_j) \end{cases}$$

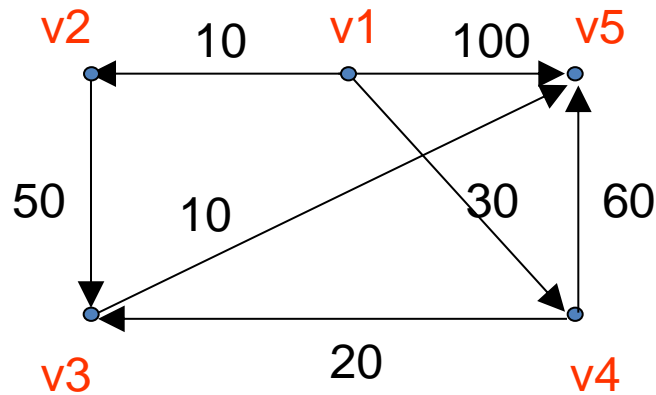
Problema do Menor Caminho Entre Dois Vértices

- Algoritmo Dijkstra -determinar:

```
S={v0};      D[v0]=0;
para cada v pertencente a V-{v0}
    D[v]=L(v0,V);
enquanto (s <> V){
    escolha o vértice w pertence V-S tal que D[w] seja mínimo
        e congele o antecessor
    coloque w em S, isto é, faça S=S U {w}
    para cada v pertencente V-S faça{
        D[v]=min(D[v],d[w]+L(w,v))
        se (d[w]+L(w,v)<D[v])
            D[v].antecessor=w;
    }
}
```


Problema do Menor Caminho Entre Dois Vértices

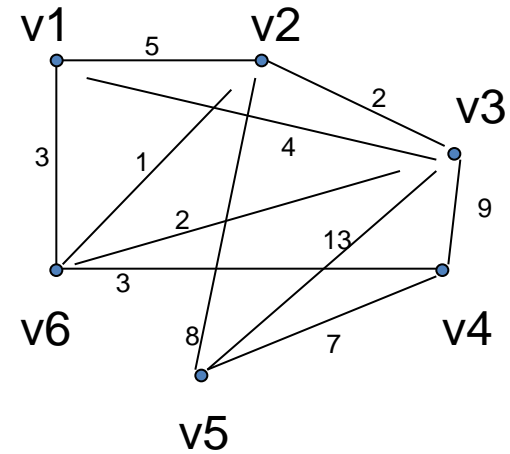
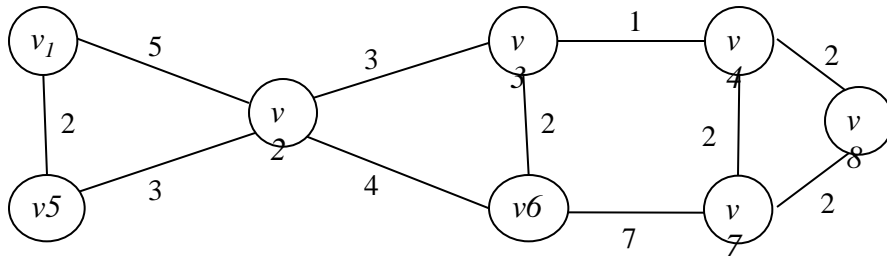
- Ex Dijkstra



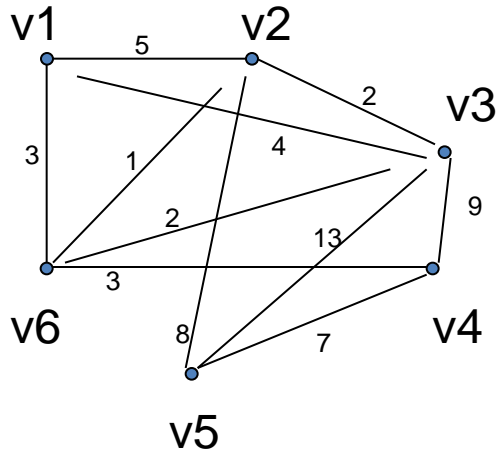
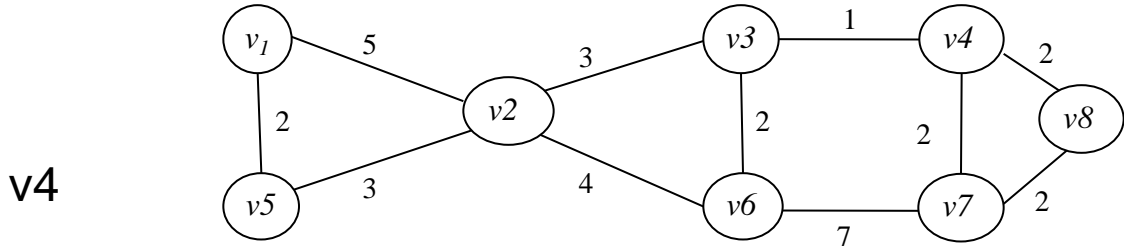
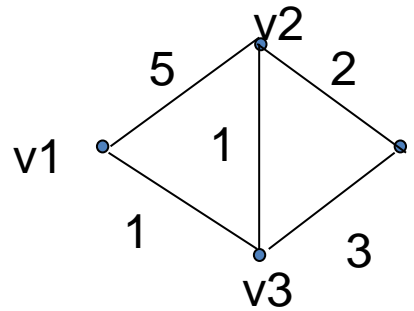
$$\begin{bmatrix} 0 & 10 & \infty & 30 & 100 \\ \infty & 0 & 50 & \infty & \infty \\ \infty & \infty & 0 & \infty & 10 \\ \infty & \infty & 20 & 0 & 60 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix}$$

Iteração	S	W	d(w)	d[2]	d[3]	d[4]	d[5]
Início	v1	-	-	10(v1)	∞	30(v1)	100(v1)
1	V1,v2	v2	10	-	$\infty, 10+50$ 60(v2)	30, 10+ ∞ 30(v1)	100, 10+ ∞ 100(v1)
2	V1,v2,v4	v4	30	-	60, 30+20 50(v4).limp	-	100, 30+60 90(v4)
3	V1,v2,v4,v3	v3	50	-	-	-	90, 50+10
4	V1,v2,v4,v3,v5	v5	60	-	-	-	60(v3)

Exercícios: Encontre o menor caminho entre um vértice origem até todos os outros vértices. Use Algoritmo do Dijkstra



Exercícios: Encontre um ciclo Euleriano para os grafos abaixo. Use Algoritmo do carteiro chinês



Dúvidas

