# Grafos – Parte 2

# Detecção de ciclos

- Uma grande quantidade de aplicações depende da identificação do grafo possuir ou não um ciclo
- O problema de determinar a existência de ciclos é o custo computacional envolvido
- Usa-se busca em profundidade para a detecção de ciclos

# Detecção de ciclos

- Para grafos não dirigidos o processo é simples, apesar de recursivo

```
detecao(v)
  num(v) = i++
  para todo vertice u, adjacente a v, faça
    se num(u) =0
      então anexe aresta uv ao conjunto de arestas
            detecao(u)
      senão se aresta vu não está no conjunto de arestas
            ciclo foi detectado
```

# Detecção de ciclos

- Para dígrafos o processo é um pouco mais complexo

```
detecao(v)
  num(v) = i++
  para todo vertice u, adjacente a v, faça
    se num(u) =0
       então pred(u) = v
              detecao(u)
       senão se num(u) não é infinito
                então pred(u) = v
                       ciclo foi detectado
```

# Ordenação topológica

- Aplicável a dígrafos sem ciclos, em que se queira estabelecer relações de precedência entre os vértices

- A solução para estabelecer uma ordem topológica entre os vértices é bastante simples

# Ordenação topológica

- Basta partir de um vértice com grau de entrada mínimo k (igual a 0)

- Retire todos os vértices com grau de entrada igual a k (e suas arestas) do grafo, identificando-os na ordenação

- Repita essas operações enquanto existirem vértices no grafo

# Biconectividade

- Um grafo é biconexo se a remoção de qualquer vértice não gerar vértices ou arestas desconexas

- Sua aplicabilidade inclui determinar rotas alternativas em sistemas de tráfego (físico ou mesmo de mensagens)

# Pontos de articulação

- Caso o grafo não seja biconexo, então o(s) vértice(s) cuja retirada cause a desconexão é (são) chamado(s) de pontos de articulação
- A identificação de pontos de articulação (ou arestas de cruzamento) permite determinar pontos críticos no grafo (pensando-se em problemas de fluxo)

# Problema do casamento

- Para grafos bipartidos o problema de fazer o emparelhamento entre os vértices dos dois grupos é importante

- Aqui o problema é identificar pares de vértices cujas arestas serão “mantidas”, ou de outro modo, que recursos seriam alocados a que consumidores, p. ex., caracterizando problemas de atribuição

# Emparelhamentos

- Emparelhamento máximo é aquele com o máximo número possível de arestas
- Caminho alternante é aquele em que arestas vizinhas pertencem ou não ao emparelhamento
- Um emparelhamento é máximo se não existem arestas no caminho alternante que possam ser adicionadas a ele

# Emparelhamentos

- O número de emparelhamento de um grafo é o tamanho de seu máximo emparelhamento

- Um emparelhamento perfeito é aquele em que todos os vértices do grafo estão incluídos no emparelhamento

# Emparelhamento máximo

- A determinação de emparelhamento máximo é feita com o uso de caminhos alternantes

- O que se faz é buscar arestas unindo vértices que não façam parte do emparelhamento (não casados) e adicionar tais arestas e vértices ao emparelhamento

# Problema da atribuição

- Em várias situações se quer determinar a melhor atribuição entre elementos de dois subconjuntos, minimizando custos ou maximizando lucros

- Essa é, na realidade, uma forma diferenciada de emparelhamento, com custos associados às arestas

# Problema da atribuição

- O algoritmo para esse tipo de problema (Kuhn) se baseia na eliminação sucessiva de linhas e colunas da matriz de adjacências

- O processo trabalha de acordo com a eliminação de Gauss, procurando identificar a atribuição de custo ótimo