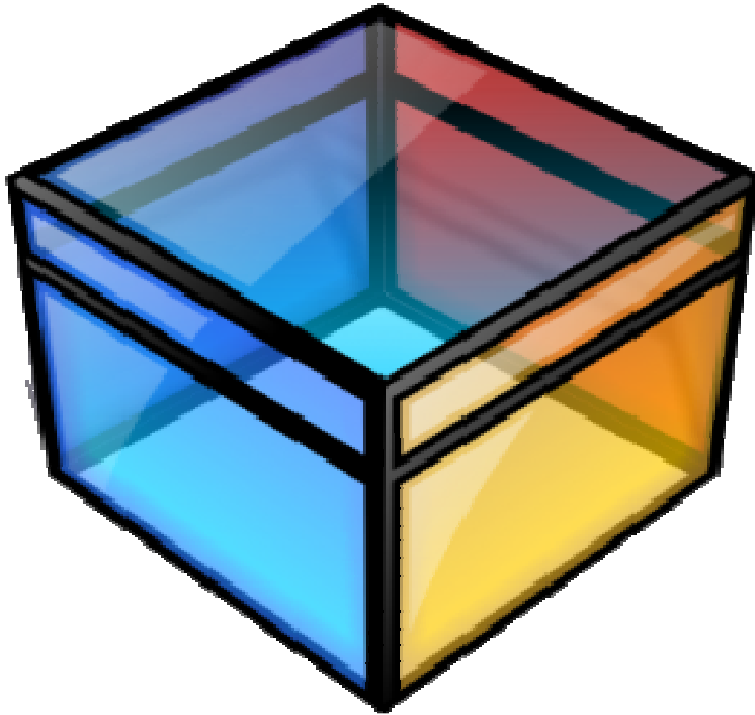


Algoritmos em Grafos



COM11087-Tópicos Especiais em Programação I
edmar.kampke@ufes.br

- Teoria dos Grafos é o estudo das propriedades e estruturas dos grafos.
- O objetivo é, após modelar um problema com grafos, identificar os principais conceitos para resolução do problema e aplicar o algoritmo ideal.
- Inicialmente vamos estudar os principais conceitos para em seguida apresentar os algoritmos

- Grafo não-direcionado conexo e acíclico
- Vértices folhas possuem grau 1
- Toda árvore possui $n-1$ arestas
- Toda árvore possui no mínimo 2 vértices folhas
- Eliminando um vértice que é folha, obtém-se outra árvore.
- Eliminando um vértice que não é folha, obtém-se um grafo desconexo

Árvores Enraizadas



- Grafo direcionado conexo e sem ciclos
- Existe apenas um vértice com grau de entrada igual a 0 (Vértice Raiz)
- Os demais possuem grau de entrada exatamente igual a 1
- Árvore binárias são exemplos de árvores enraizadas onde todos os vértices possuem grau de saída 0 ou 2
- Métade dos vértices são folhas

Árvores Geradoras



- Subgrafo gerador (possui todos os vértices) e conexo que contenha $n-1$ arestas
- Todo grafo conexo possui pelo menos uma árvore geradora
- BFS e DFS constroem um árvore geradora
- Existem algoritmos para encontrar a árvore geradora mínima (com menor custo)

- Grafo conexo com apenas 1 caminho entre cada par de vértices
- A existência de árvore geradora prova que um grafo é conexo (usar BFS e DFS)
- Articulação (biconexo = grafo sem Articulação) / Ponte
- Algoritmo de força-bruta
- Grafos direcionados diferentes níveis de conectividade

- Grafo conexo que não é árvore possui ciclos
- Ciclo Euleriano: Achar um caminho que começa em um vértice v , passa por todas as arestas apenas uma única vez e termina em v .
- Para ter Ciclo Euleriano: Todos os vértices devem ter grau par
- Construção de um Ciclo Euleriano: Usar DFS e achar um ciclo, eliminar o ciclo no grafo original, achar outro ciclo e fazer o merge até ter n vértices isolados.

- Ciclo Hamiltoniano: Achar um caminho que começa em um vértice v , passa por todos os vértices apenas uma única vez e termina em v .
- Problema do Caixeiro Viajante (em Grafo Valorado)
- Não existe algoritmo eficiente para determinar Ciclo Hamiltoniano para grafos grandes
- Grafos pequenos pode-se usar *Backtracking*

- Grafo Planar: Grafo que pode ser desenhado no plano sem cruzamento de arestas
- Árvores são sempre planares
- Diversos problemas (construção de redes e estradas)
- $n - m = f + 2$ $m \geq 3n - 2$ (Condição)
- K_5 e $K_{3,3}$ não são planares

Árvore Geradora Mínima



- Muito útil para definir o menor grafo que conecta todos os vértices
- Exemplo: Número mínimo de estradas conectando cidades
- Algoritmos: Prim e Kruskal
- Prim é apresentado por ser facilmente adaptado ao problema de caminhos mínimos.

Árvore Geradora Mínima



➤ Definição

O peso de um grafo valorado é a soma dos pesos das arestas do grafo

➤ Definição

A árvore geradora mínima é a árvore geradora de peso mínimo

Árvore Geradora Mínima



➤ Algoritmo de PRIM para construção de uma árvore geradora mínima

$v \leftarrow$ um vértice qualquer de G

$T_1 \leftarrow \{v\}$

Para $k = 1, \dots, (|V|-1)$

$e_k \leftarrow$ uma aresta* (u,v) **de peso mínimo** com

$u \in T_k$ e $v \notin T_k$

*se houver mais de uma escolha qualquer uma delas

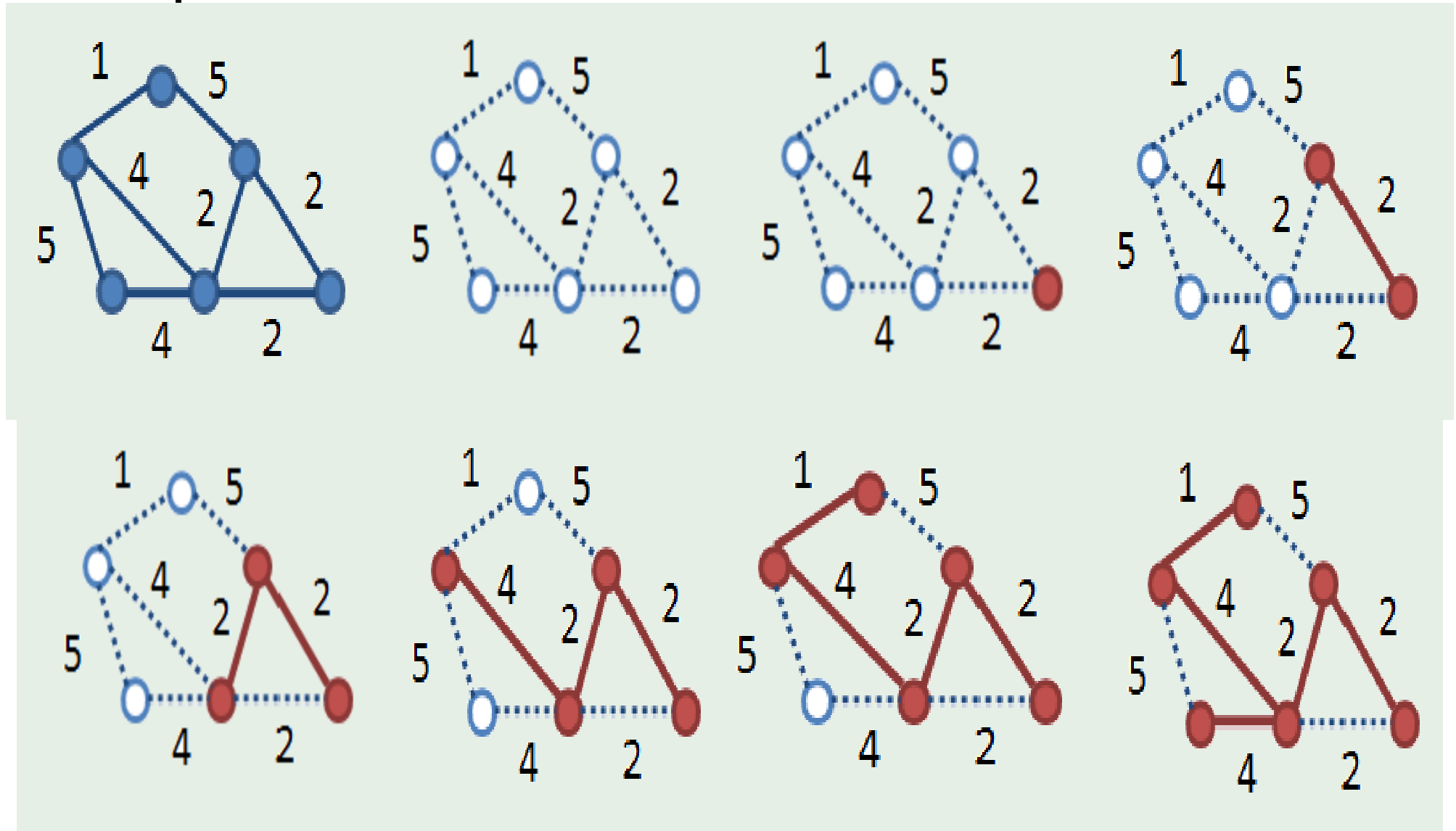
$T_{k+1} \leftarrow T_k \cup e_k + v$

Retornar $T_{|V|}$

Árvore Geradora



➤ Exemplo



Árvore Geradora Mínima



- Outros problemas com AGM:
 - Árvore Geradora Máxima
 - Árvore Geradora de Produto Mínimo
 - Árvore Geradora com Gargalo Mínimo

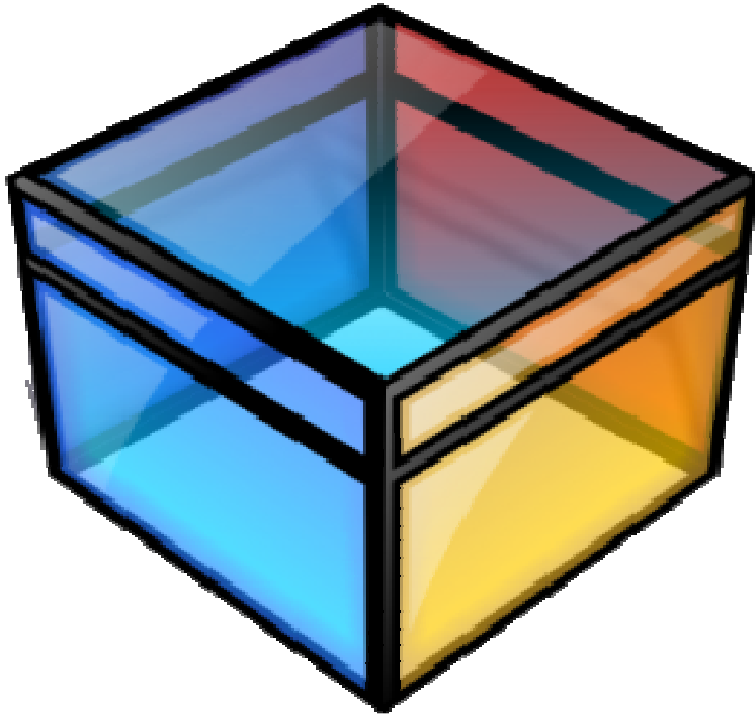
- Dado um grafo valorado o objetivo é calcular o menor caminho entre um par de vértices
- Algoritmo de Dijkstra é muito conhecido pois calcula a distância de um vértice para todos os demais
- Pequenas alterações no algoritmo de Prim podem ser feitas
- Algoritmo de Floyd calcula o menor caminho entre todos os pares de vértices

- Um grafo direcionado, chama se grafo de fluxo se ele possui um (e apenas um) vértice s que possui grau de entrada igual a 0 e um (e apenas um) vértice t que possui grau de saída igual a 0
- O objetivo é enviar o maior fluxo de s para t obedecendo as restrições de capacidade dos arcos e de contenção de fluxo
- Um dos algoritmos mais conhecidos é o de Ford-Fulkerson

- Encontrar o fluxo máximo em um grafo de fluxo pode ser útil em vários problemas, principalmente em problemas de Matching em grafos bipartidos
- Exemplos:
 - Problema de Alocação de trabalhadores
 - Problema do Casamento
 - Problema dos dois processadores e subrotinas

- *The Grand Dinner*
- *Freckles*
- *Railroads*
- *War*
- *Tourist Guide*
- *The Problem With the Problem Setter*

Algoritmos em Grafos



COM11087-Tópicos Especiais em Programação I
edmar.kampke@ufes.br