Estrutura de Dados II

Árvore Geradora Mínima

Algoritmo de Kruskal

Algoritmo Genérico

- A cada passo adicionamos a S uma aresta (u, v) que não viola o invariante. (u, v) é chamada de uma aresta segura.

```
    void GenericoAGM
1       S = ∅;
2       while (S não constitui uma árvore geradora mínima)
3           (u, v) = seleciona (A);
4           if (aresta (u, v) é segura para S) S = S + {(u, v)}
5       return S;
```

Algoritmo de Kruskal

- **Pode ser derivado do algoritmo genérico.**
- **S é uma floresta e a aresta segura adicionada a S é sempre uma aresta de menor peso que conecta dois componentes distintos.**
- **Considera as arestas ordenadas pelo peso.**

Algoritmo de Kruskal

- **Este algoritmo encontra a aresta segura (e com menos peso) procurando em todas as arestas que conectam quaisquer duas árvores na floresta A**
  - A cada passo o algoritmo de Kruskal adiciona à floresta a aresta com menor peso possível

- **O algoritmo de Kruskal utiliza conjuntos disjuntos, em cada conjunto contém os vértices de uma árvore na floresta atual**
  - A operação FIND-SET(u) retorna um elemento representativo do conjunto que contém u
  - Assim, para determinar se dois vértices u e v pertencem à mesma árvore, basta testar se FIND-SET(u) é igual a FIND-SET(v)
  - A combinação de árvores é executada com o procedimento UNION

Algoritmo de Kruskal

- **Versão inicial**

AGM-KRUSKAL(G, w)
1 $A \leftarrow \emptyset$
2 Ordene as arestas em ordem não-decrescente de peso
3 **para cada** $(u, v) \in E$ nessa ordem **faça**
4 **se** u e v estão em componentes distintos de (V, A)
5 **então** $A \leftarrow A \cup \{(u, v)\}$
6 **devolva** A

Problema: Como verificar eficientemente se u e v estão no mesmo componente da floresta $G_A = (V, A)$?

Algoritmo de Kruskal

Inicialmente $G_A = (V, \emptyset)$, ou seja, G_A corresponde à floresta onde cada componente é um vértice isolado.

Ao longo do algoritmo, esses componentes são modificados pela inclusão de arestas em A.

Uma estrutura de dados para representar $G_A = (V, A)$ deve ser capaz de executar eficientemente as seguintes operações:

- Dado um vértice u, determinar o componente de G_A que contém u e
- dados dois vértices u e v em componentes distintos C e C', fazer a união desses em um novo componente.

Estrutura de Dados para conjuntos disjuntos

Uma estrutura de dados para conjuntos disjuntos deve ser capaz de executar as seguintes operações:

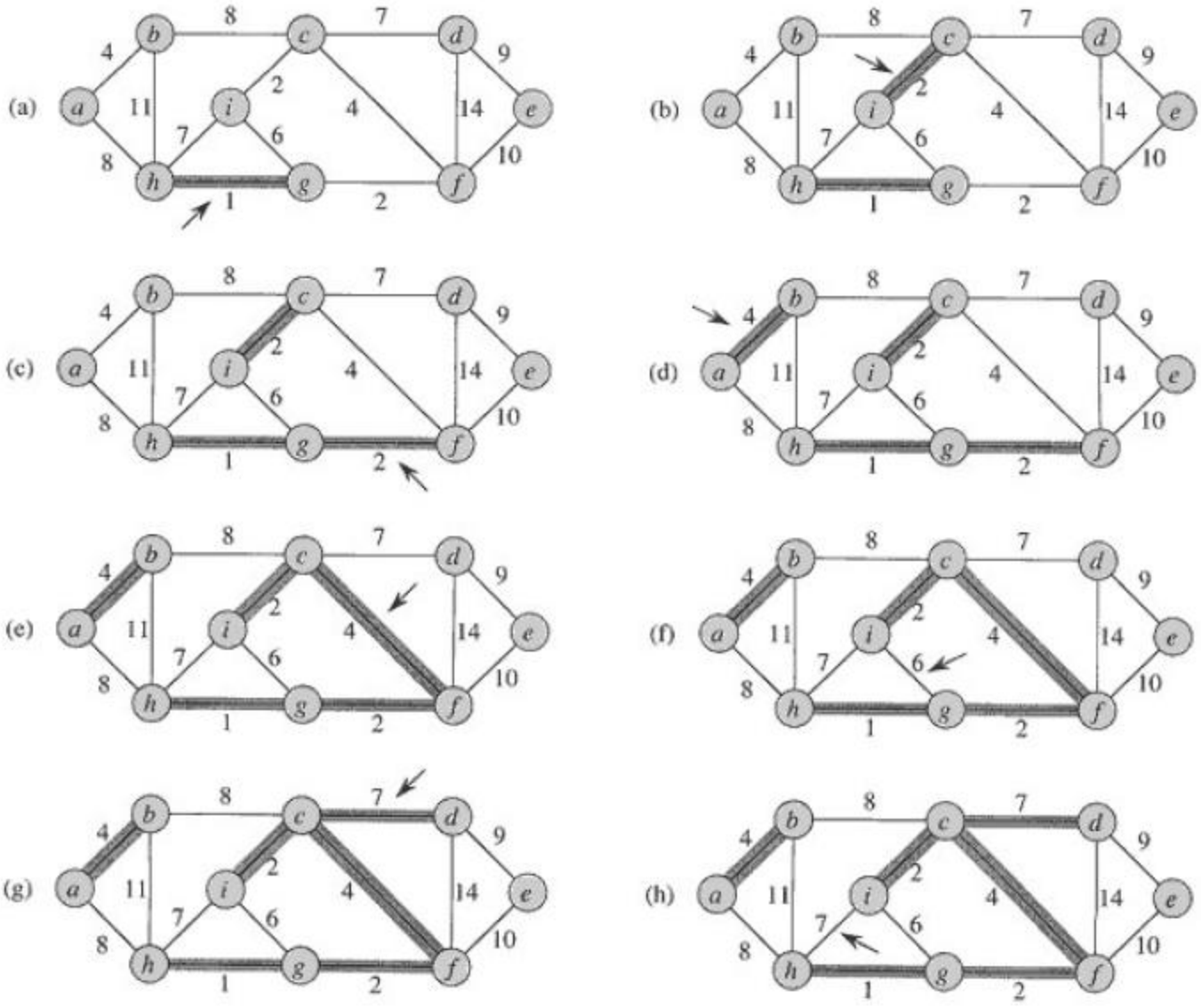
- MAKE-SET(x): cria um novo conjunto $\{x\}$.
- UNION(x, y): une os conjuntos (disjuntos) que contém x e y, digamos S_x e S_y, em um novo conjunto $S_x \cup S_y$.

  Os conjuntos S_x e S_y são descartados da coleção.
- FIND-SET(x) devolve um apontador para o representante do (único) conjunto que contém x.

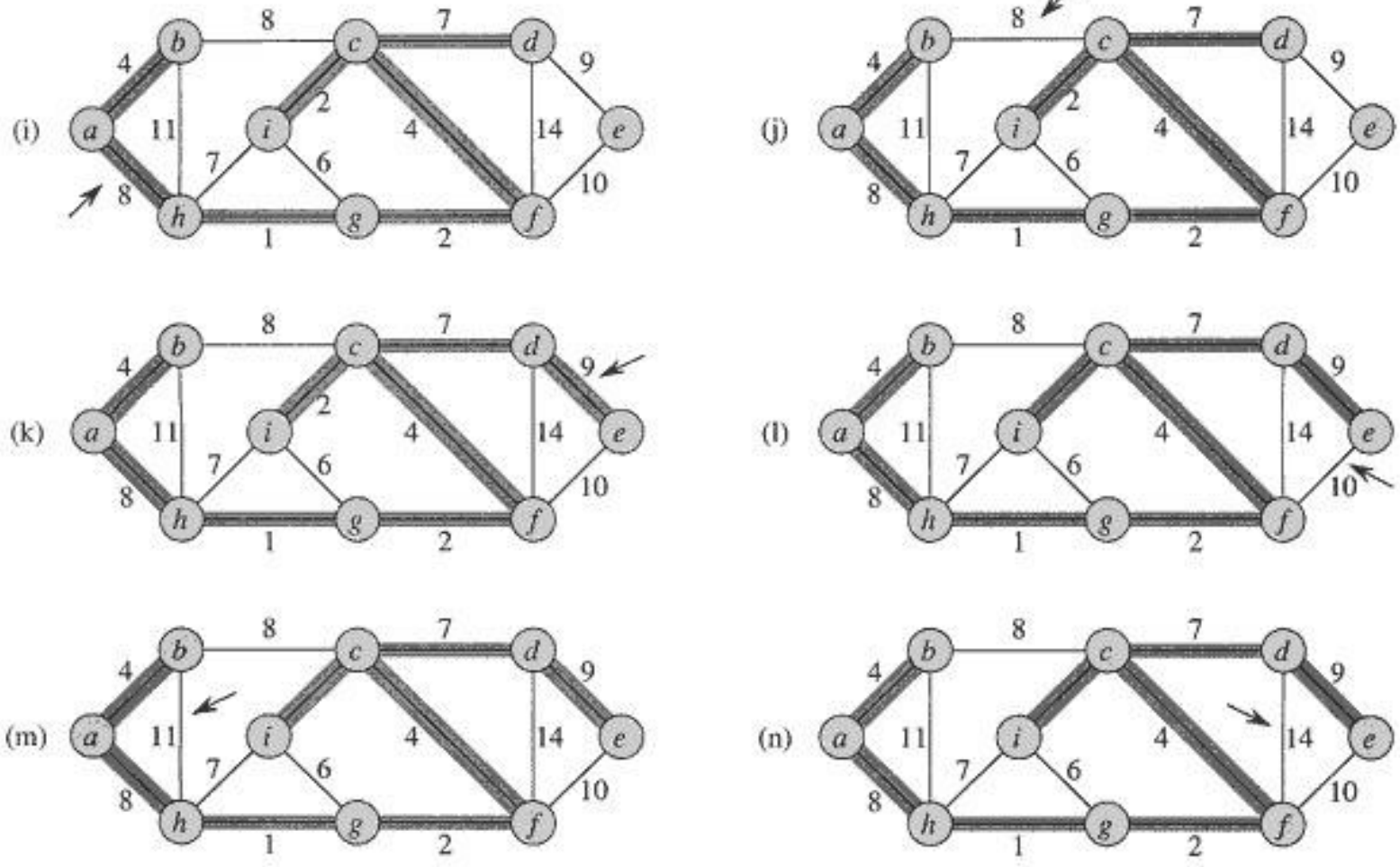
Algoritmo de Kruskal

```
MST-KRUSKAL(G, w)
1  A ← ∅
2  for each vertex v ∈ V[G]
3       do MAKE-SET(v)
4  sort the edges of E into nondecreasing order by weight w
5  for each edge (u, v) ∈ E, taken in nondecreasing order by weight
6       do if FIND-SET(u) ≠ FIND-SET(v)
7            then A ← A ∪ {(u, v)}
8                 UNION(u, v)
9  return A
```

Algoritmo de Kruskal



Algoritmo de Kruskal



Análise Algoritmo de Kruskal

- O tempo de execução do algoritmo de Kruskal sobre um grafo G = (V, E) depende da implementação da estrutura de dados conjuntos-disjuntos
  - Inicialização consome O(|V|)
  - Ordenação consome O(|E| lg |E|)
  - Se for utilizada a estrutura de dados UNION-FIND com a heurística "compressão-de-caminho" (Cormen, capítulo 21), cada operação sobre o conjunto-disjunto é da ordem O(lg |E|)

- Portanto, o tempo computacional do algoritmo de Kruskal é O(|E| lg |E|)

Referencias

- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. Algoritmos: Teoria e Prática. Editora Campus, 2002
- Ziviani, N. Projeto de Algoritmos Com Implementações em Pascal e C, Cengage Learning, 2004.
- Notas de aula. Prof. Rafael Fernandes DAI/IFMA
- Notas de aula. Profa. Leticia Bueno UFABC
- **http://www.facom.ufu.br/~madriana/EBD/Didatica.pdf**
- **http://www.ic.unicamp.br/~zanoni/mo417/2011/aulas/handout/11-arvore-geradora-minima.pdf**