Data Structure - Spring 2022

17. Graph – Part 3

Walid Abdullah Al
Computer and Electronic Systems Engineering
Hankuk University of Foreign Studies

TA: **Seong Joo Kim**

Based on:
Goodrich, Chapter 9,11
Karumanchi, Chapter 6-7
Slides by Prof. Yung Yi, KAIST
Slides by Prof. Chansu Shin, HUFS

Computer Vision Lab
Hankuk University of Foreign Studies

Minimum Spanning Trees

Spanning subgraph

- Subgraph of a graph G containing all the vertices of G

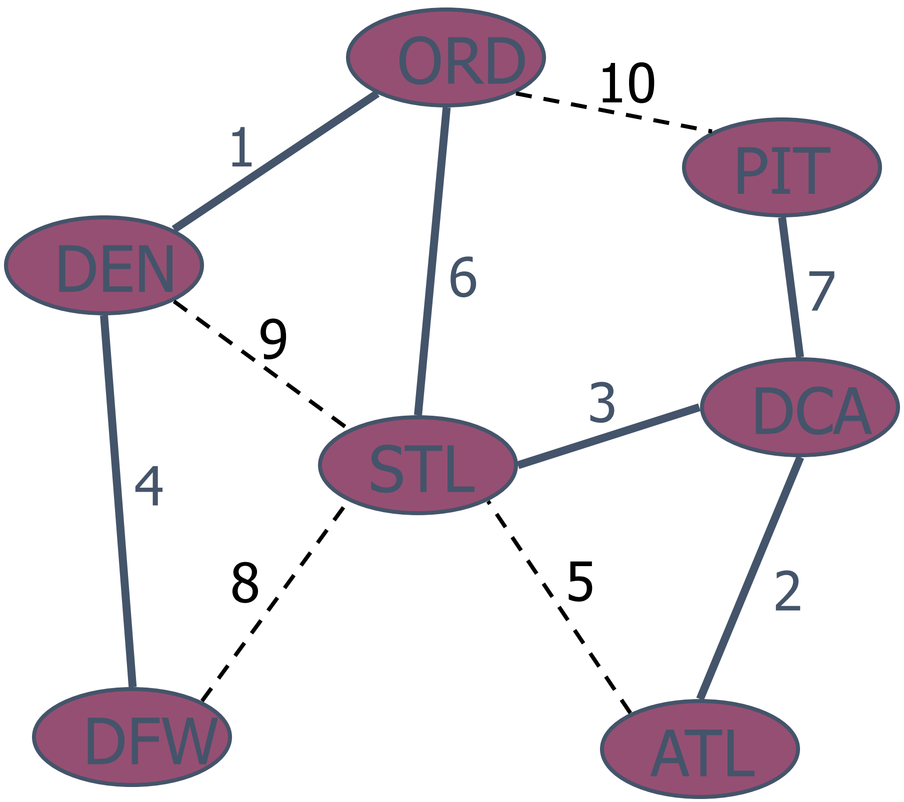
Spanning tree

- Spanning subgraph that is itself a (free) tree

Minimum spanning tree (MST)

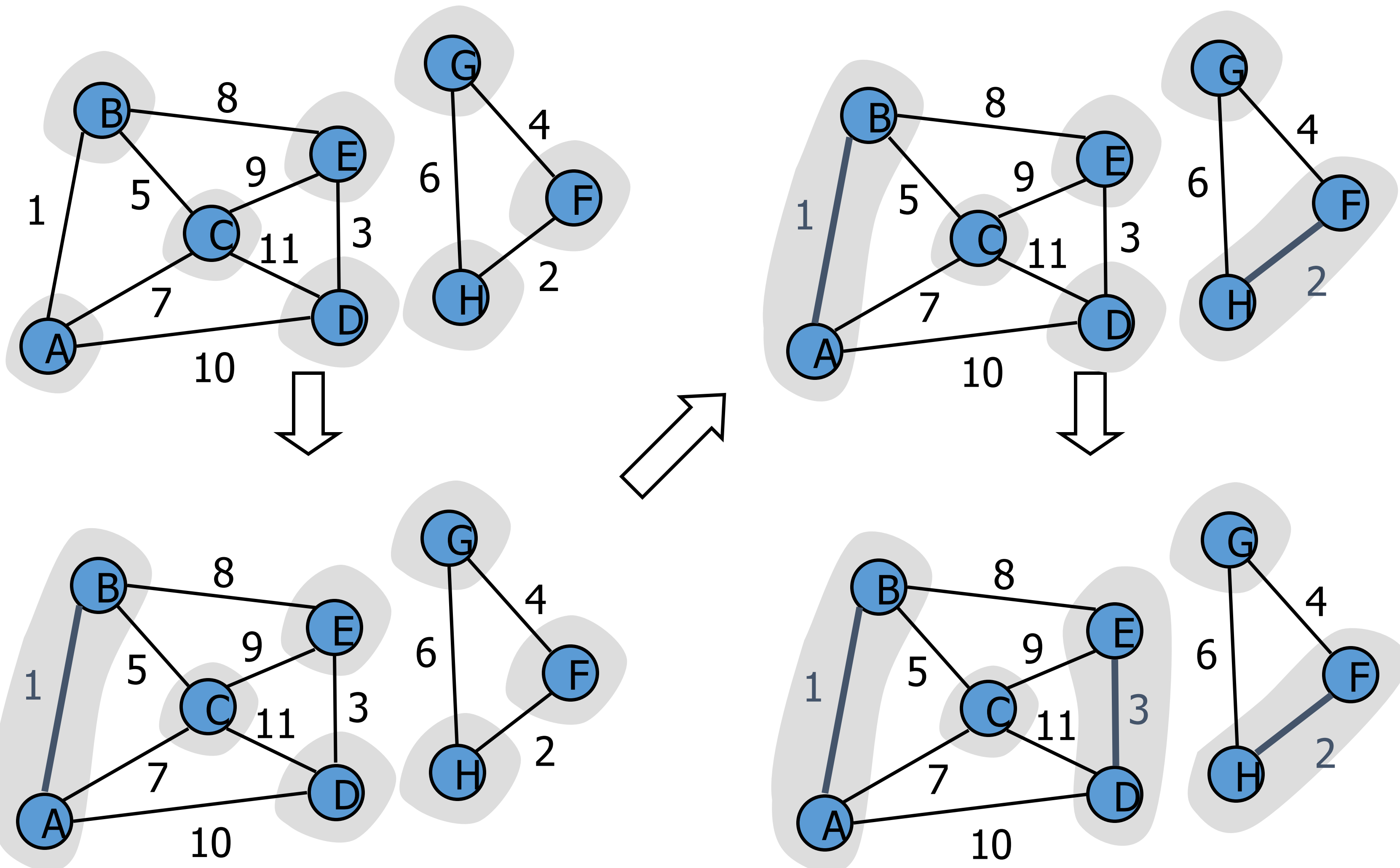
- Spanning tree of a weighted graph with minimum total edge weight

- **Applications**
  - Communications networks
  - Transportation networks

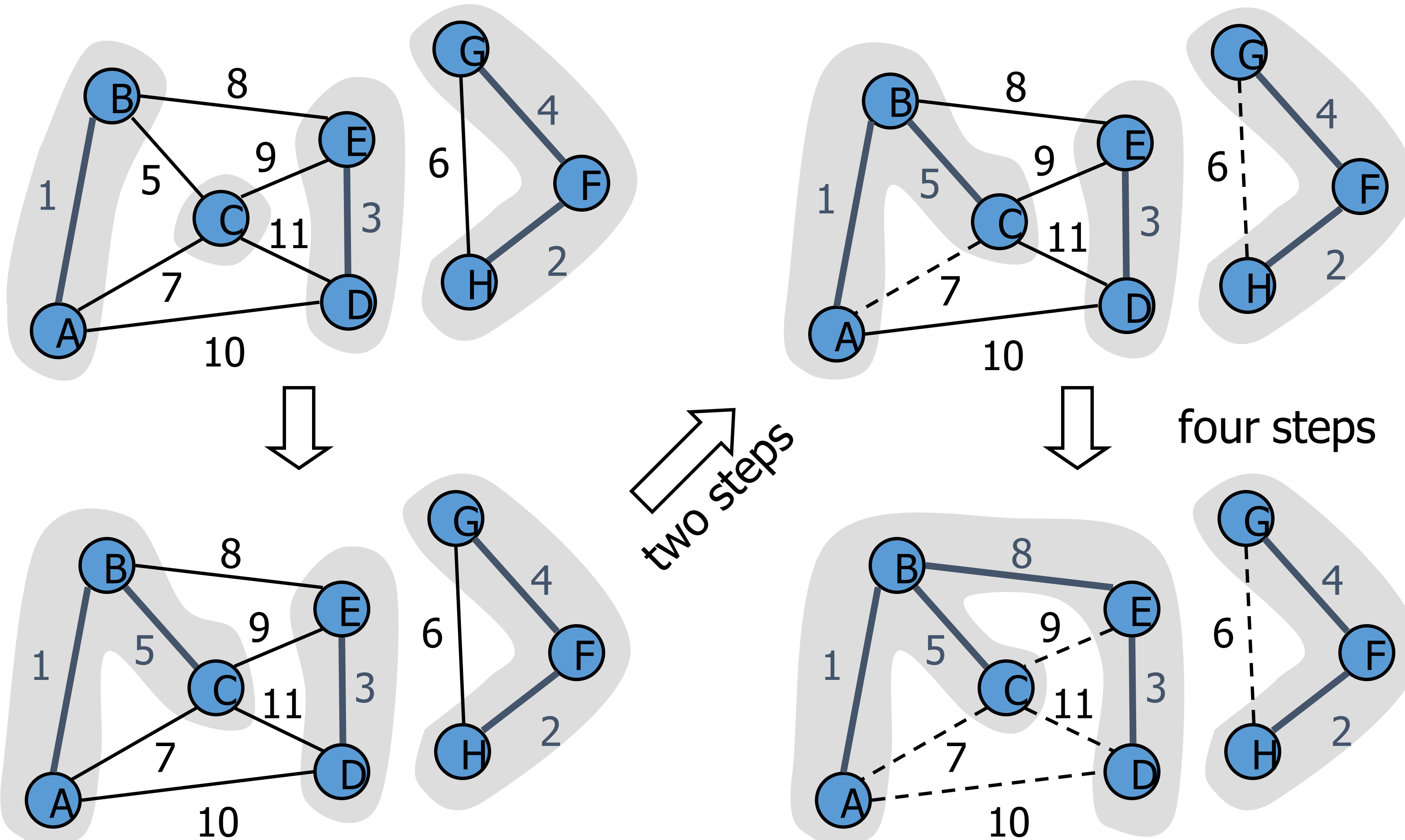


Kruskal’s Algorithm

Kruskal’s Algorithm: Example



Example (contd.)



Kruskal's Algorithm

- **Maintain a partition of the vertices into clusters**
  - Initially, single-vertex clusters
  - Keep an MST for each cluster
  - Merge "closest" clusters and their MSTs
- **A priority queue stores the edges outside clusters**
  - Key: weight
  - Element: edge
- **At the end of the algorithm**
  - One cluster and one MST (if connected)

Algorithm *KruskalMST*(*G*)
    for each vertex ***v*** in ***G*** **do**
        Create a cluster consisting of ***v***
    let ***Q*** be a priority queue.
    Insert all edges into ***Q***
    $T \leftarrow \emptyset$
    {***T*** is the union of the MSTs of the clusters}
    while ***T*** has fewer than ***n*** – 1 edges **do**
        e ← ***Q.removeMin***().***getValue***()
        [***u***, ***v***] ← ***G.endVertices***(***e***)
        A ← ***getCluster***(***u***)
        B ← ***getCluster***(***v***)
        if $A \neq B$ **then**
            Add edge ***e*** to ***T***
            mergeClusters(***A***, ***B***)
    return ***T***