CSE408
Minimum Spanning Tree(Prims,Kruskshal)

Lecture #26

Minimum Spanning Tree

- A minimum spanning tree connects all nodes in a given graph
- A MST must be a connected and undirected graph
- A MST can have weighted edges
- Multiple MSTs can exist within a given undirected graph

More about Multiple MSTs

- Multiple MSTs can be generated depending on which algorithm is used
- If you wish to have an MST start at a specific node
- However, if there are weighted edges and all weighted edges are unique, only one MST will exist

Real Life Application of a MST

A cable TV company is laying cable in a new neighborhood. If it is constrained to bury the cable only along certain paths, then there would be a graph representing which points are connected by those paths. Some of those paths might be more expensive, because they are longer, or require the cable to be buried deeper; these paths would be represented by edges with larger weights. A *minimum spanning tree* would be the network with the lowest total cost.

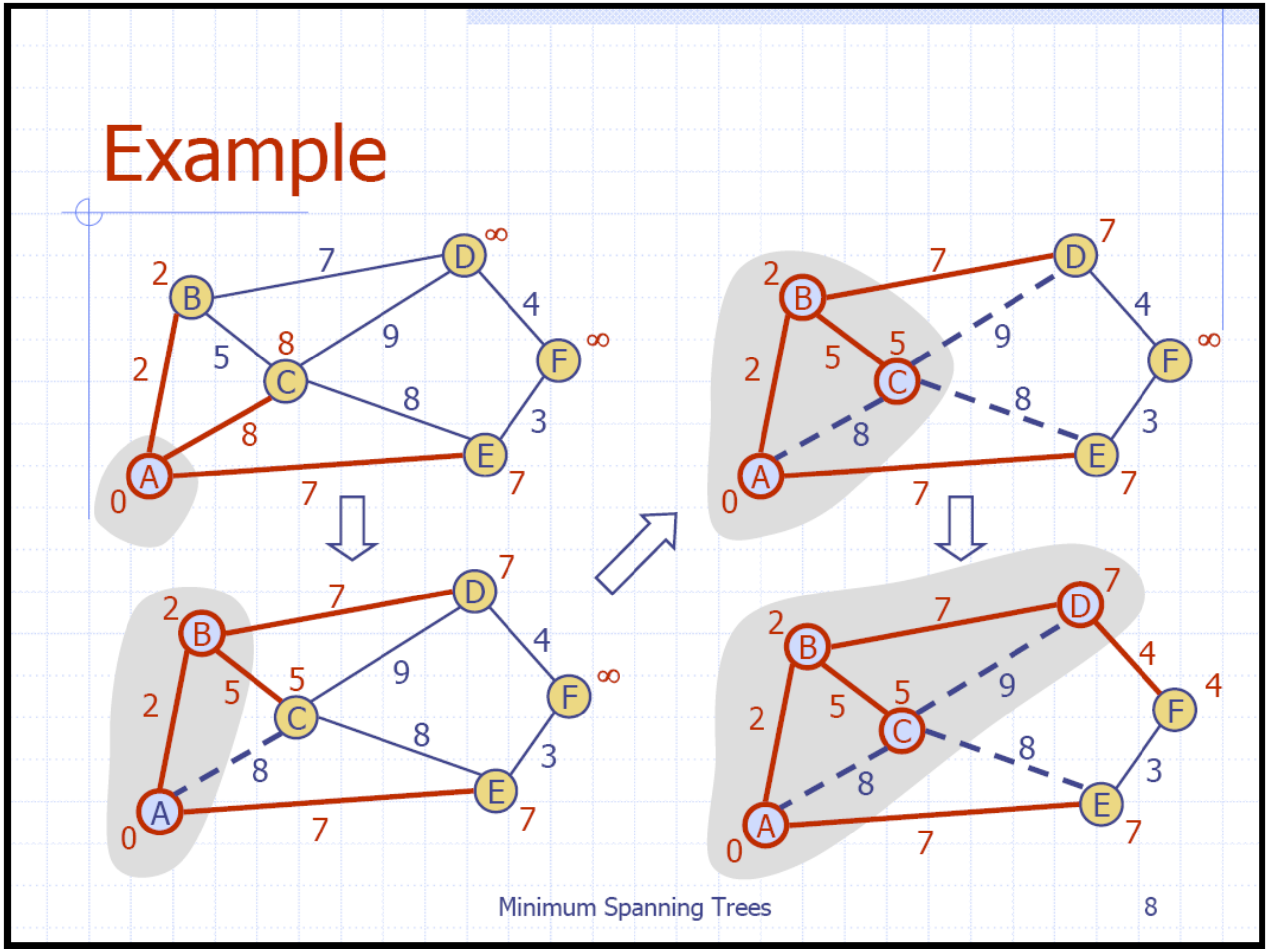
Prim’s Algorithm

- Initially discovered in 1930 by Vojtěch Jarník, then rediscovered in 1957 by Robert C. Prim
- Similar to Dijkstra’s Algorithm regarding a connected graph
- Starts off by picking any node within the graph and growing from there

Prim’s Algorithm Cont.

- Label the starting node, A, with a 0 and all others with infinite
- Starting from A, update all the connected nodes’ labels to A with their weighted edges if it less than the labeled value
- Find the next smallest label and update the corresponding connecting nodes
- Repeat until all the nodes have been visited

Prim’s Algorithm Example

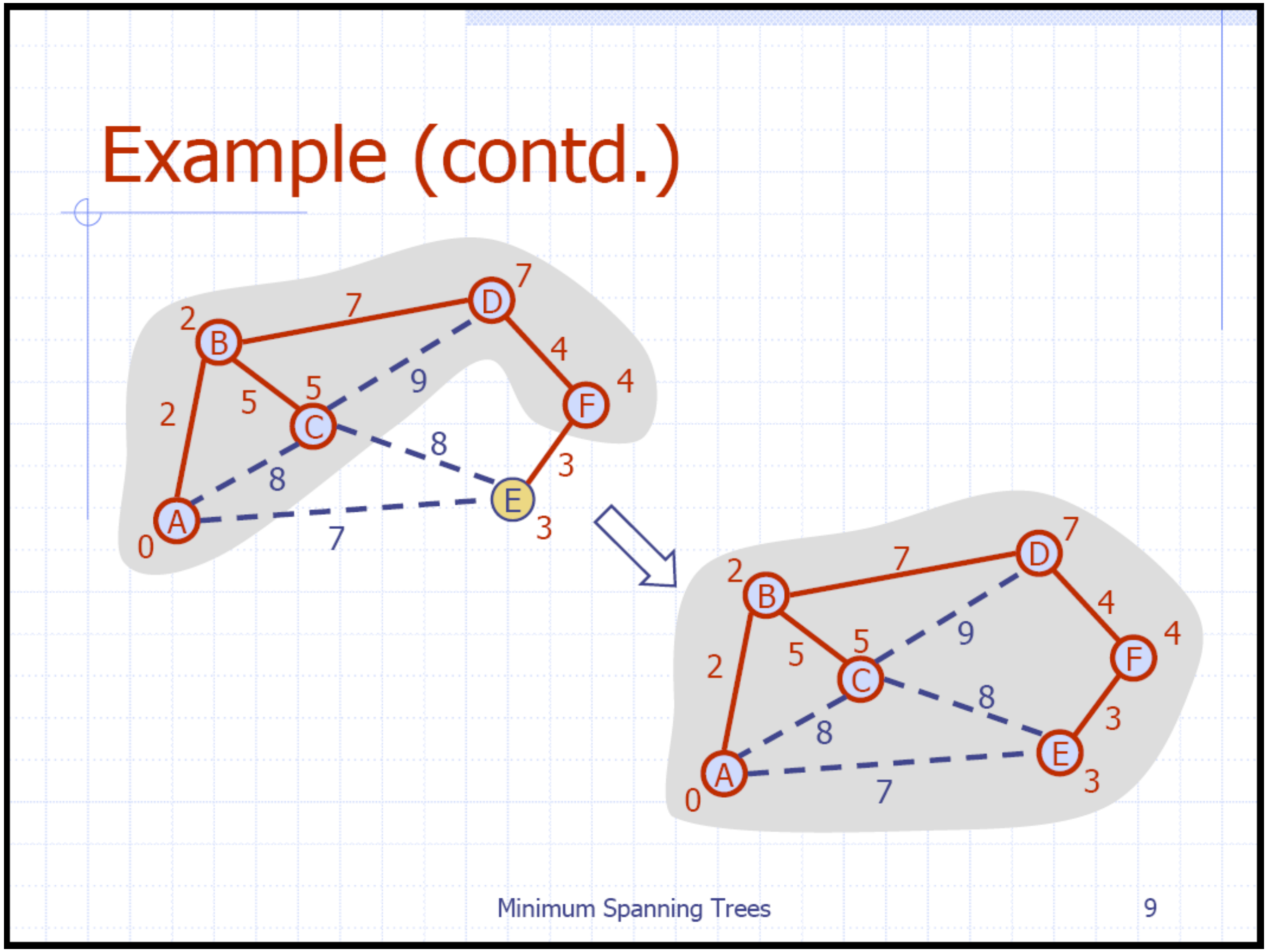


Example

Minimum Spanning Trees

8

Prim’s Algorithm Example



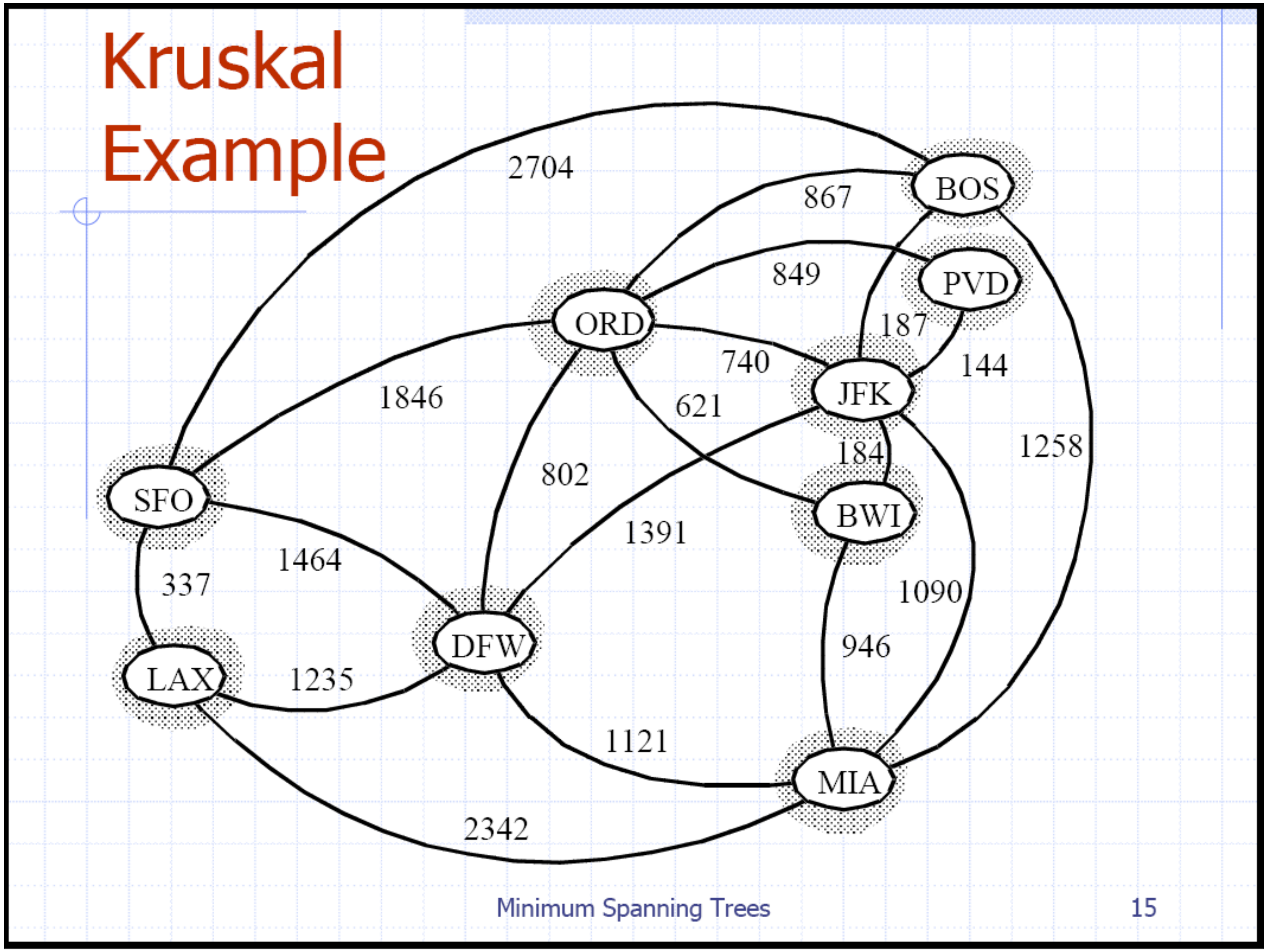
Example (contd.)

Minimum Spanning Trees

Kruskal’s Algorithm

- Created in 1957 by Joseph Kruskal
- Finds the MST by taking the smallest weight in the graph and connecting the two nodes and repeating until all nodes are connected to just one tree
- This is done by creating a priority queue using the weights as keys
- Each node starts off as it’s own tree
- While the queue is not empty, if the edge retrieved connects two trees, connect them, if not, discard it
- Once the queue is empty, you are left with the minimum spanning tree

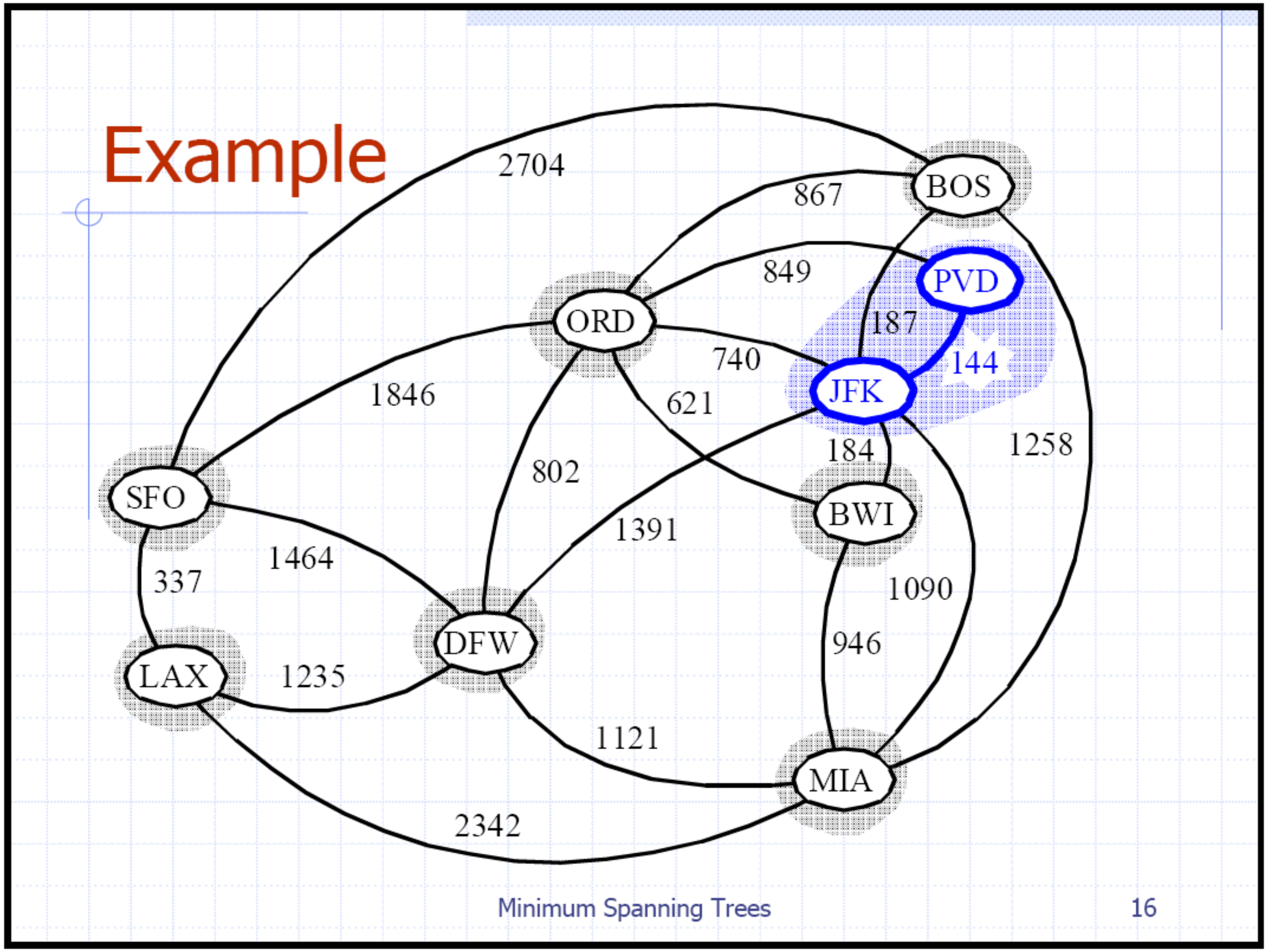
Kruskal’s Algorithm Example



Kruskal Example

Minimum Spanning Trees 15

Kruskal’s Algorithm Example

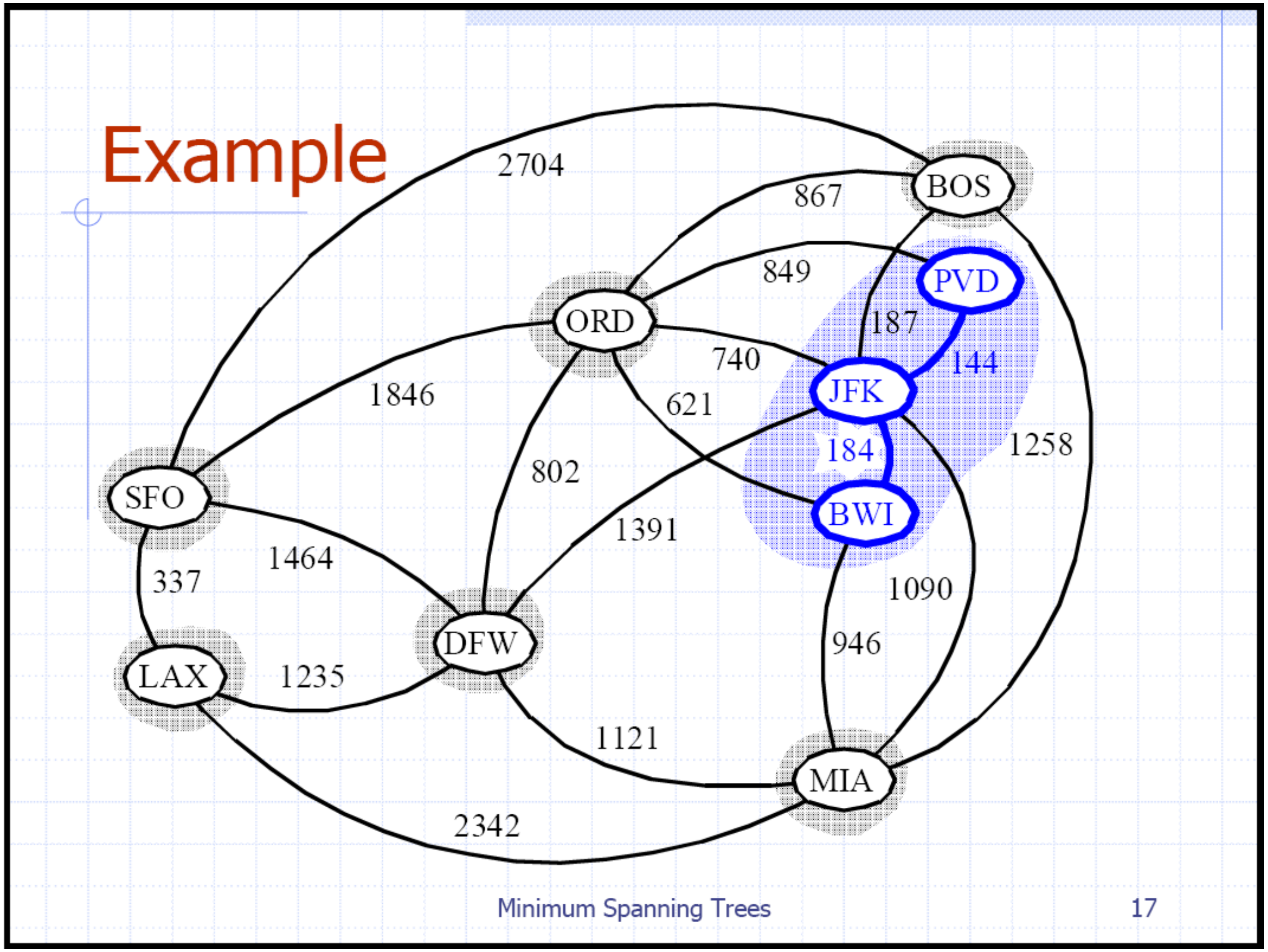


Example

Minimum Spanning Trees

16

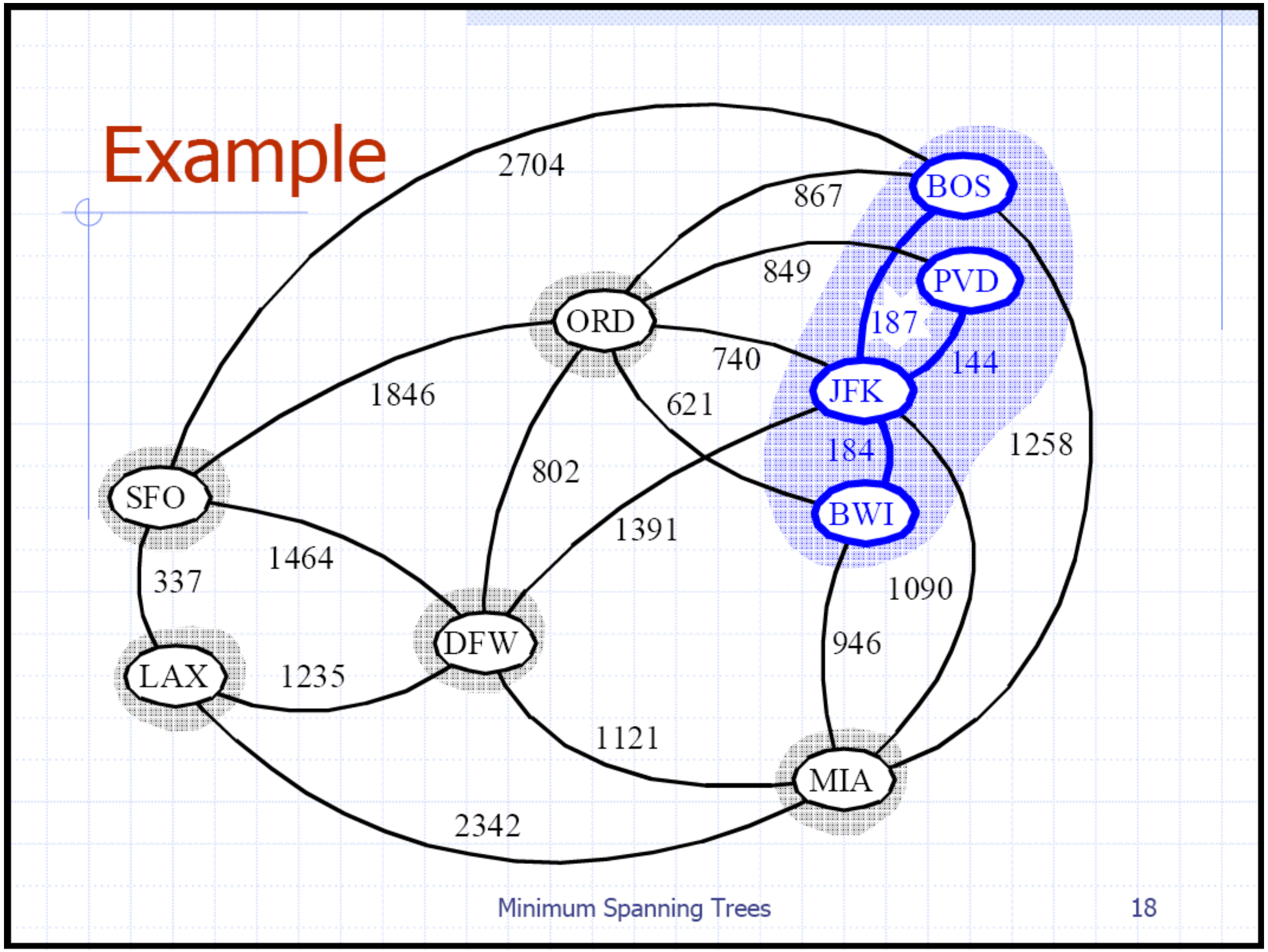
Kruskal’s Algorithm Example



Example

Minimum Spanning Trees 17

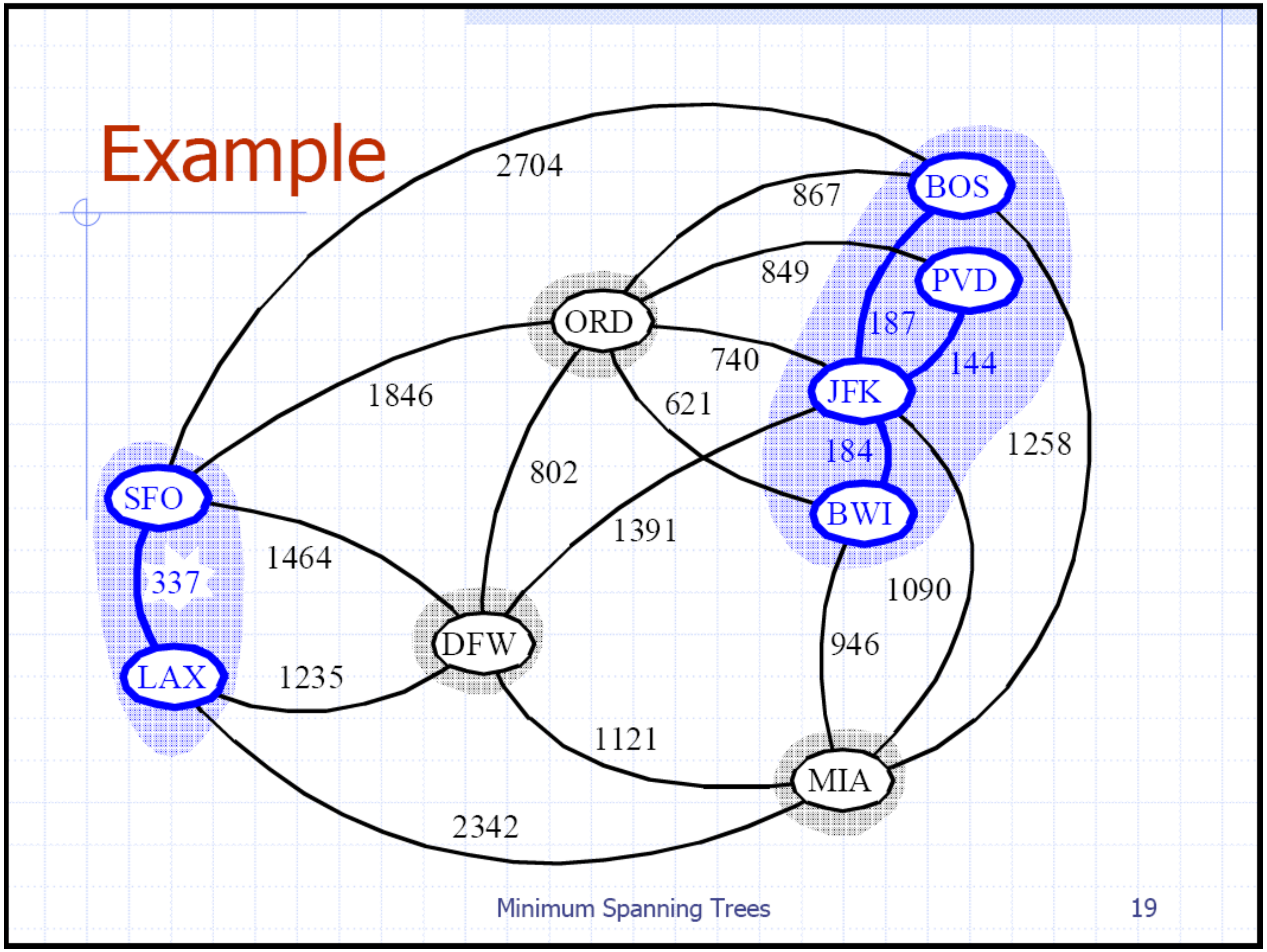
Kruskal’s Algorithm Example



Example

Minimum Spanning Trees 18

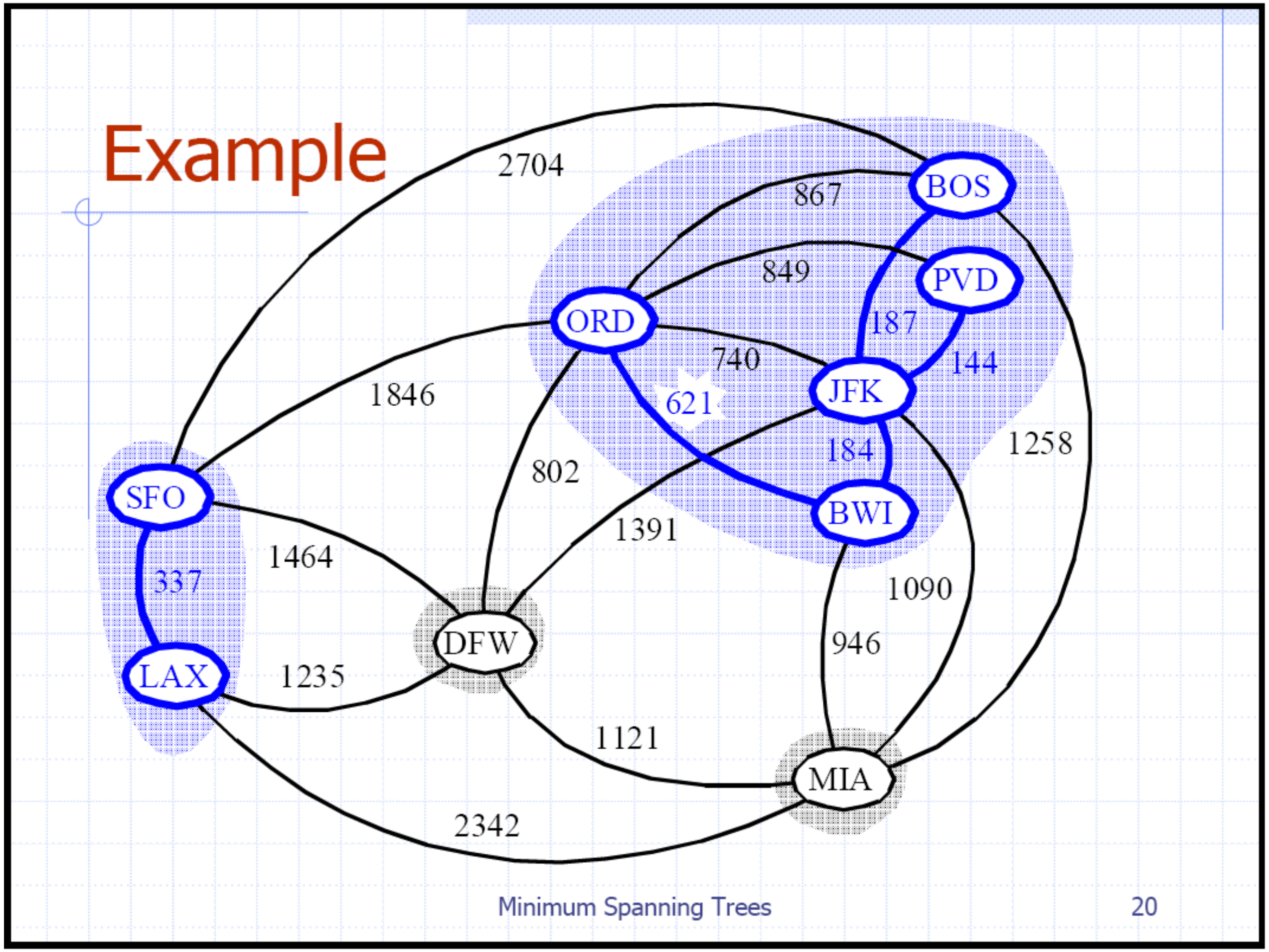
Kruskal’s Algorithm Example



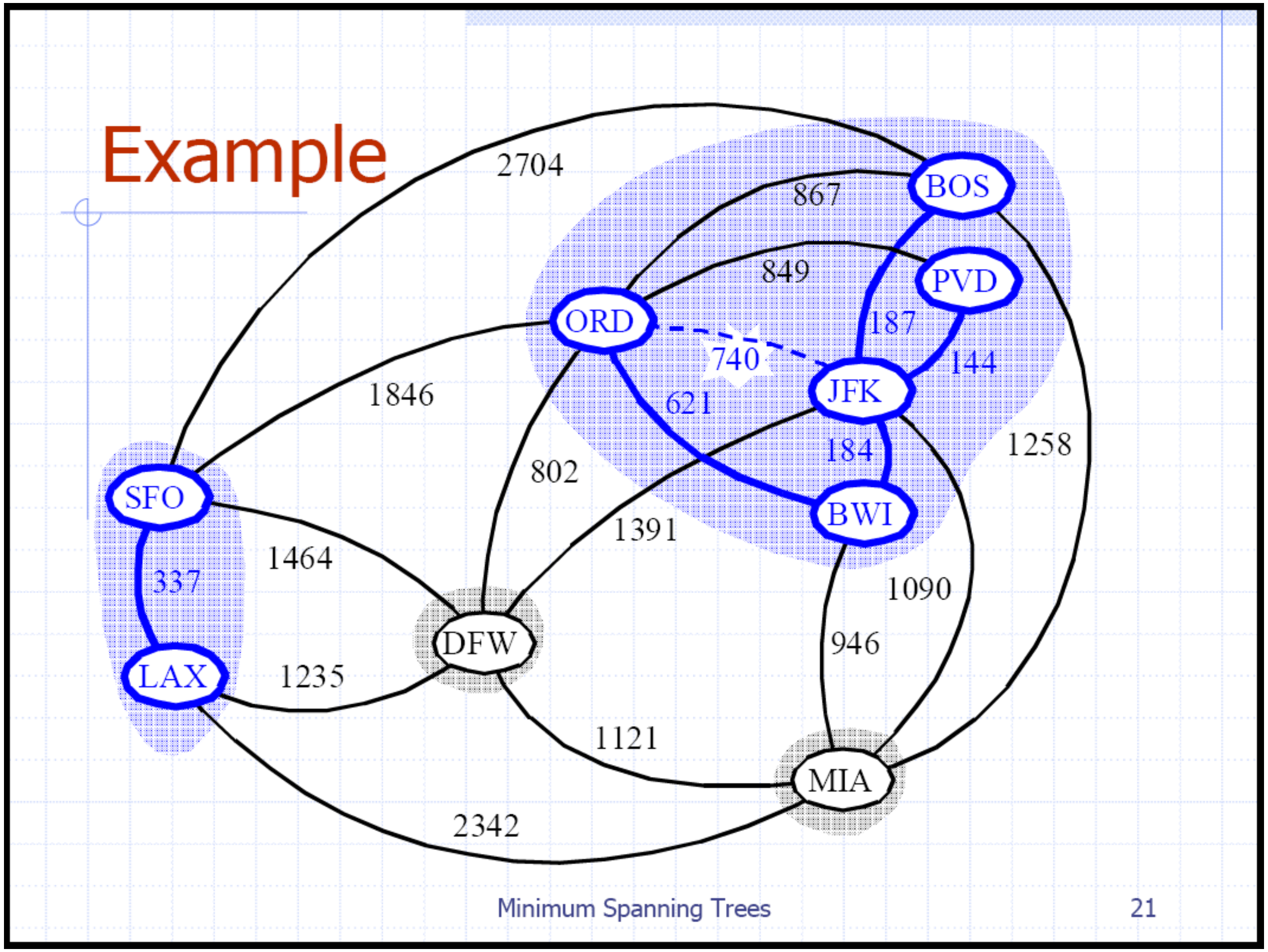
Example

Minimum Spanning Trees 19

Kruskal’s Algorithm Example



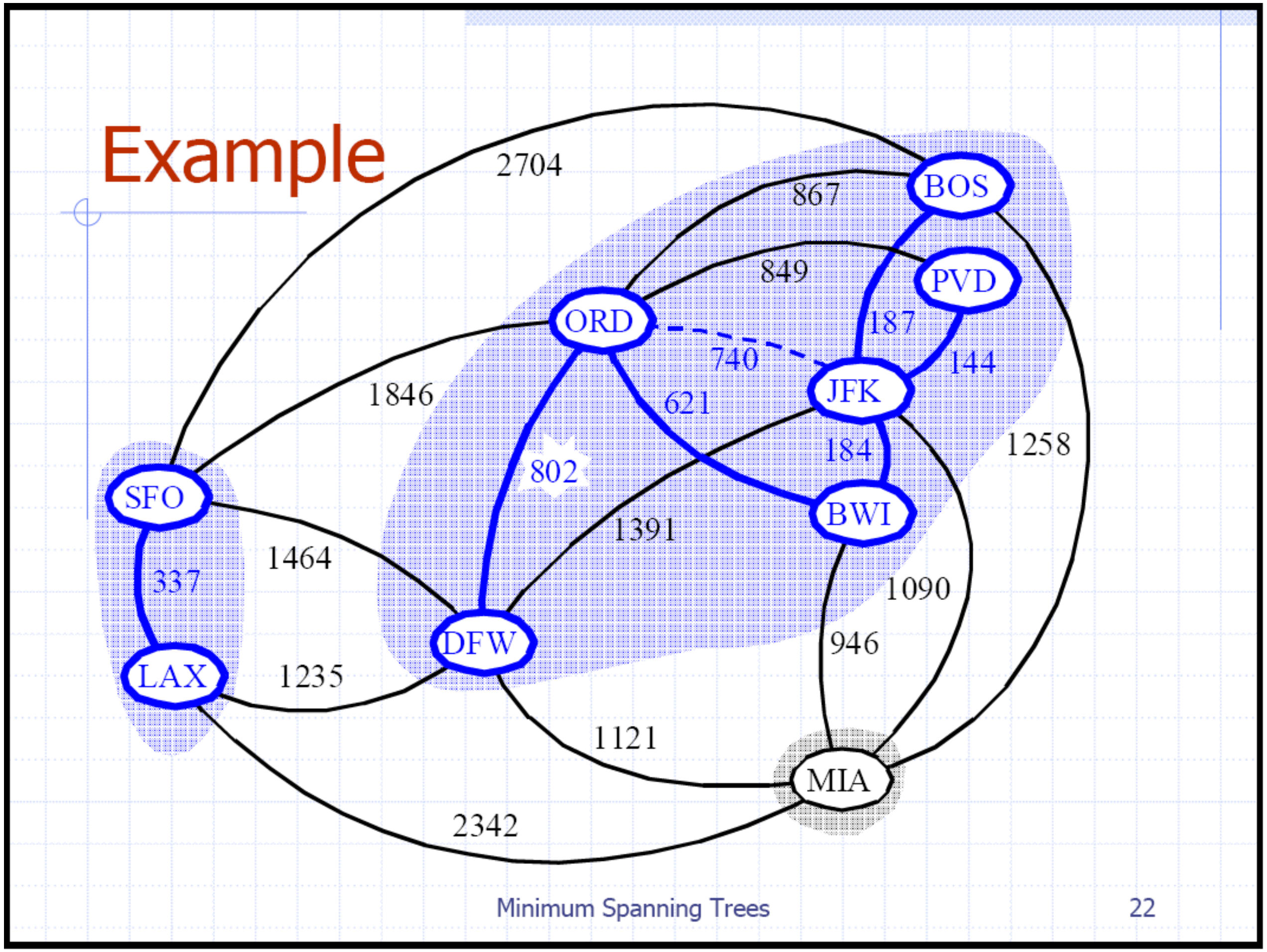
Kruskal’s Algorithm Example



Example

Minimum Spanning Trees

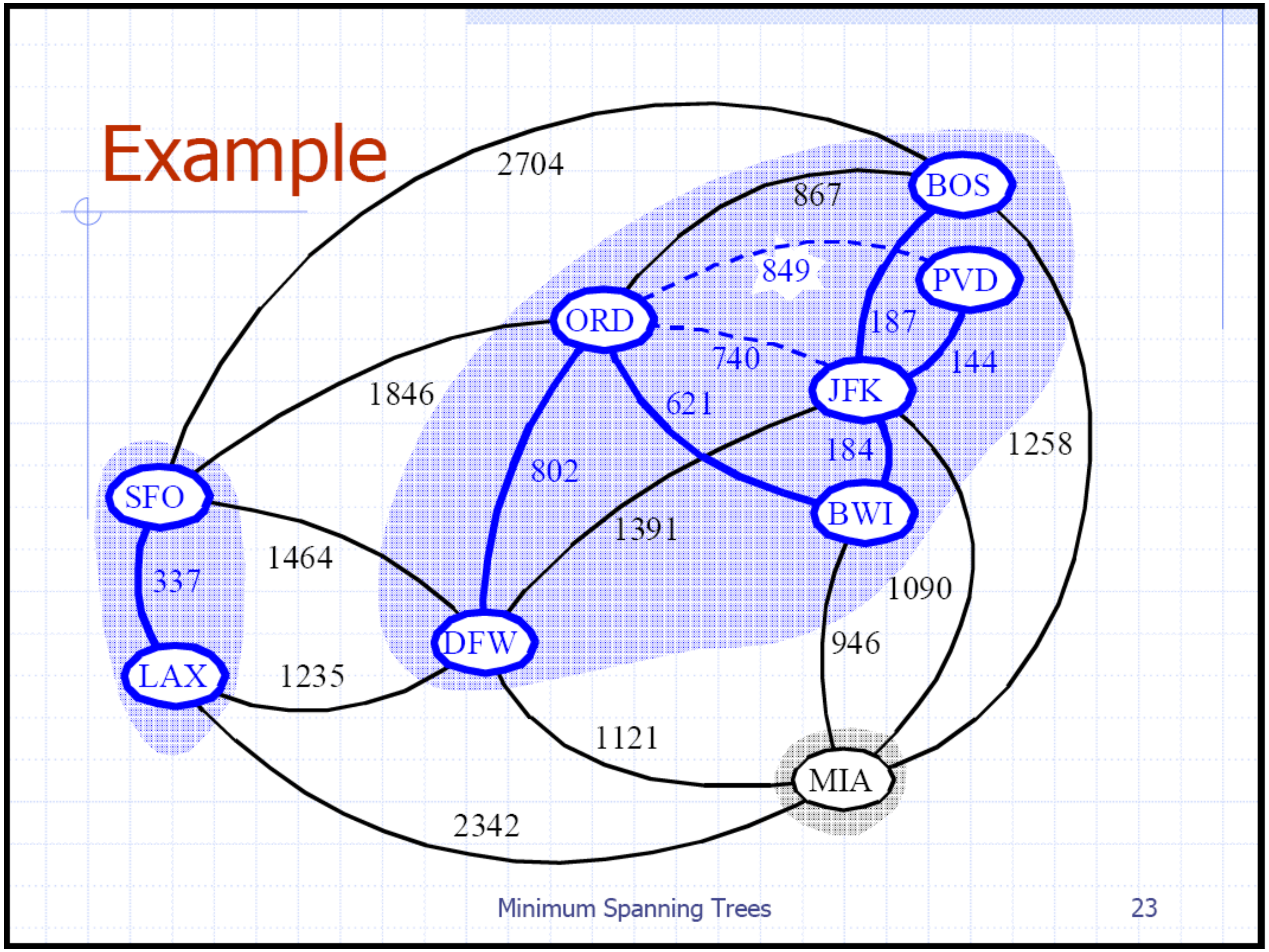
Kruskal’s Algorithm Example



Example

Minimum Spanning Trees 22

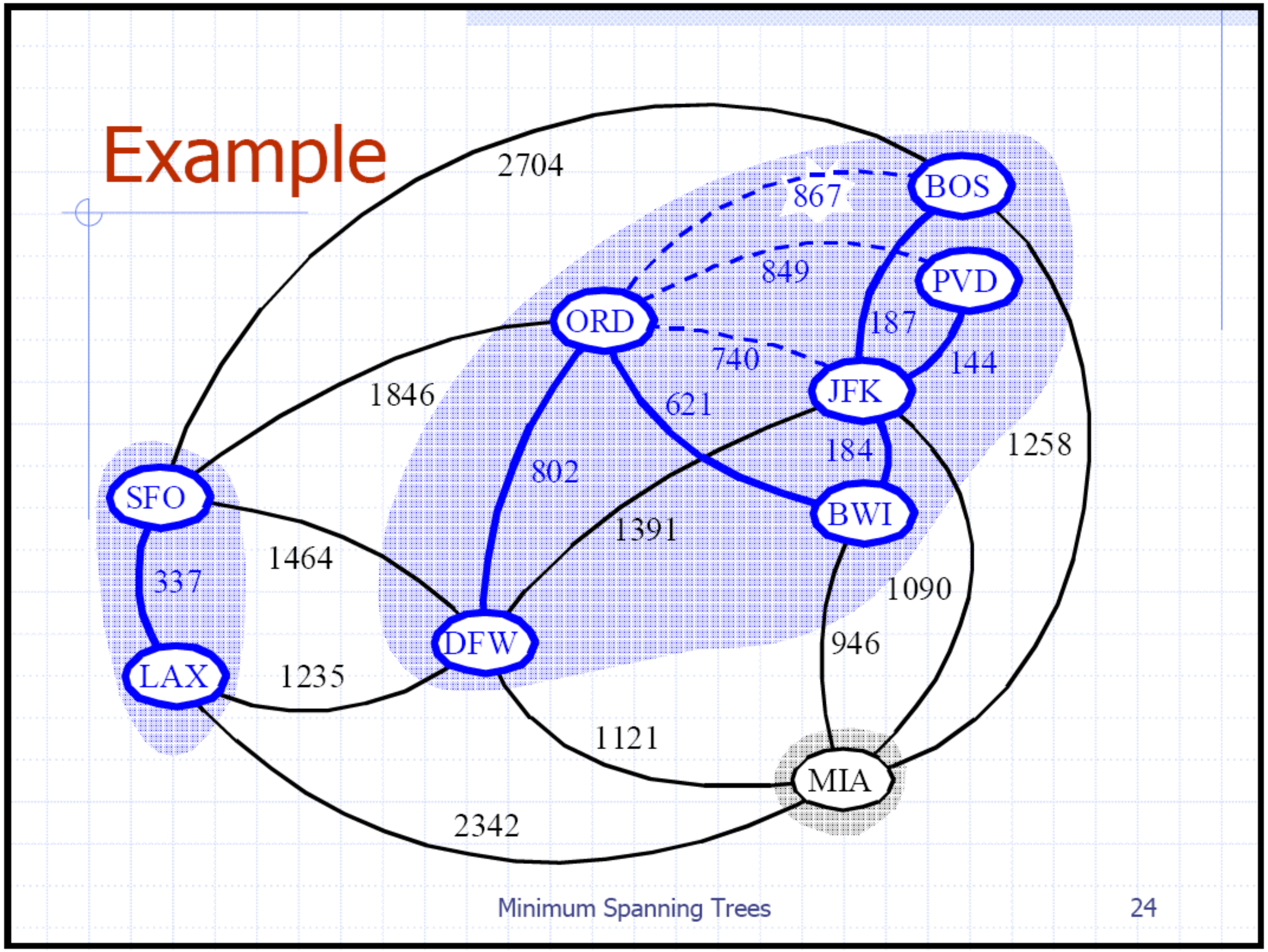
Kruskal’s Algorithm Example



Example

Minimum Spanning Trees

Kruskal’s Algorithm Example

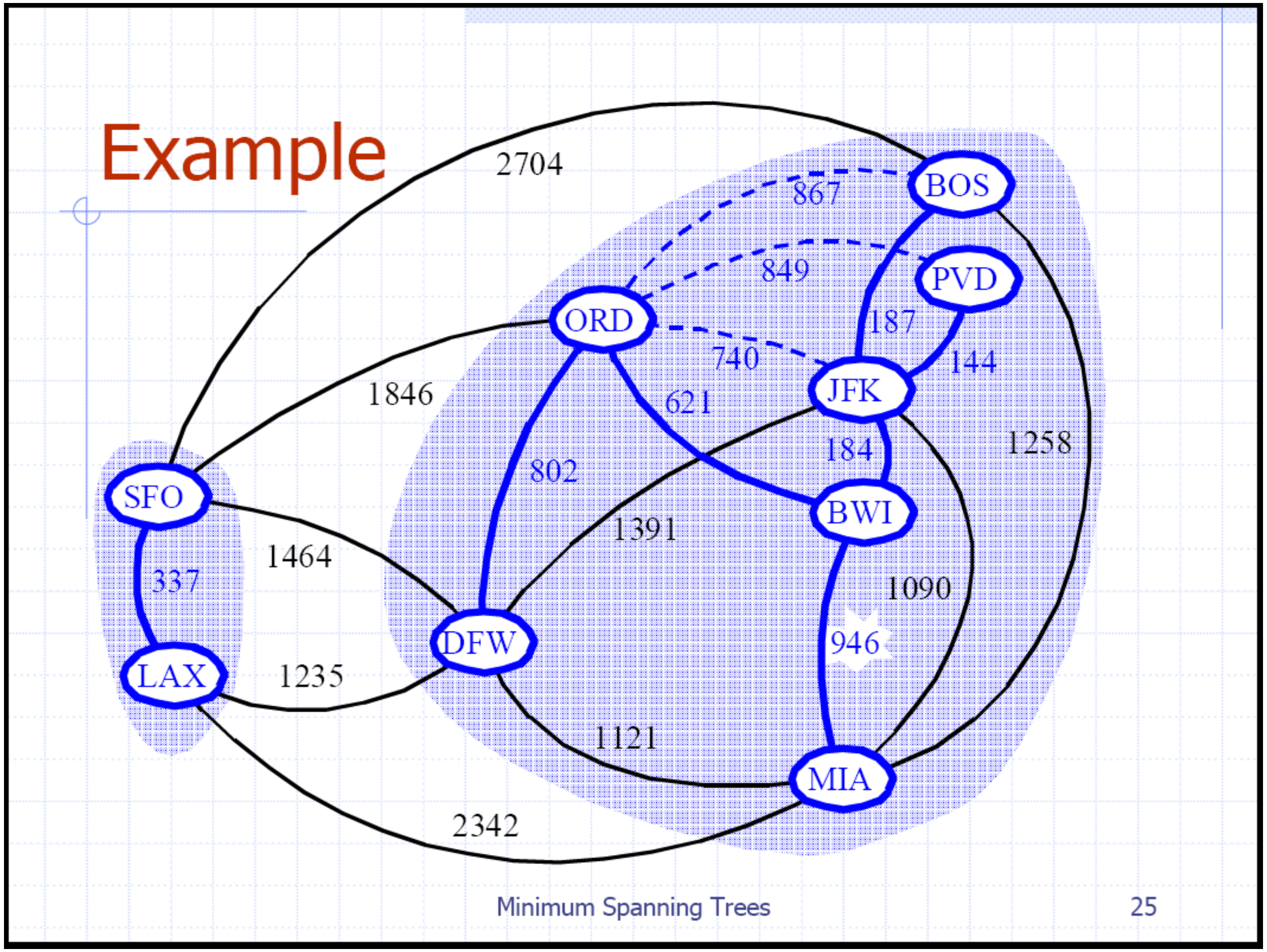


Example

Minimum Spanning Trees

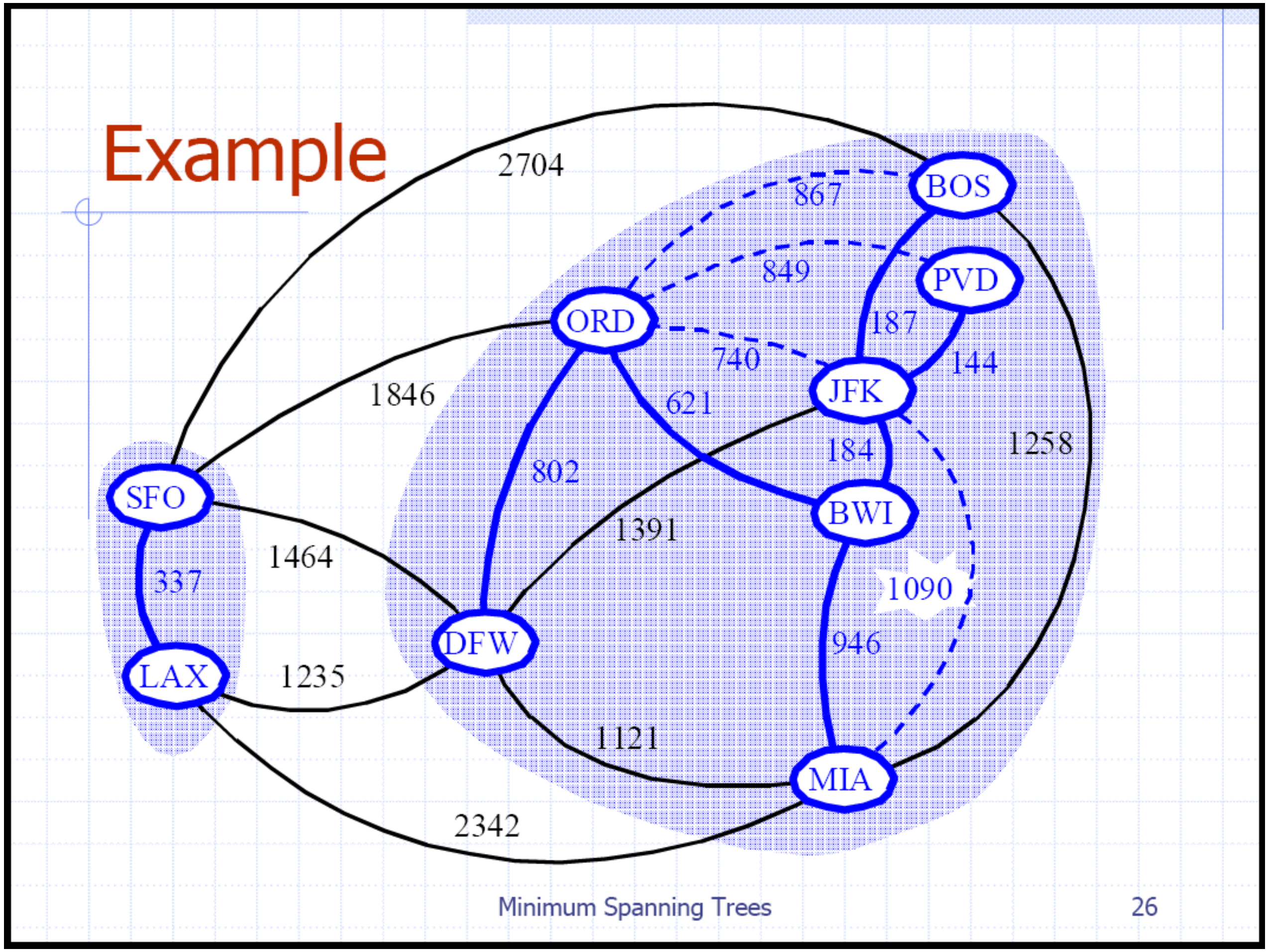
24

Kruskal’s Algorithm Example



Example

Kruskal’s Algorithm Example

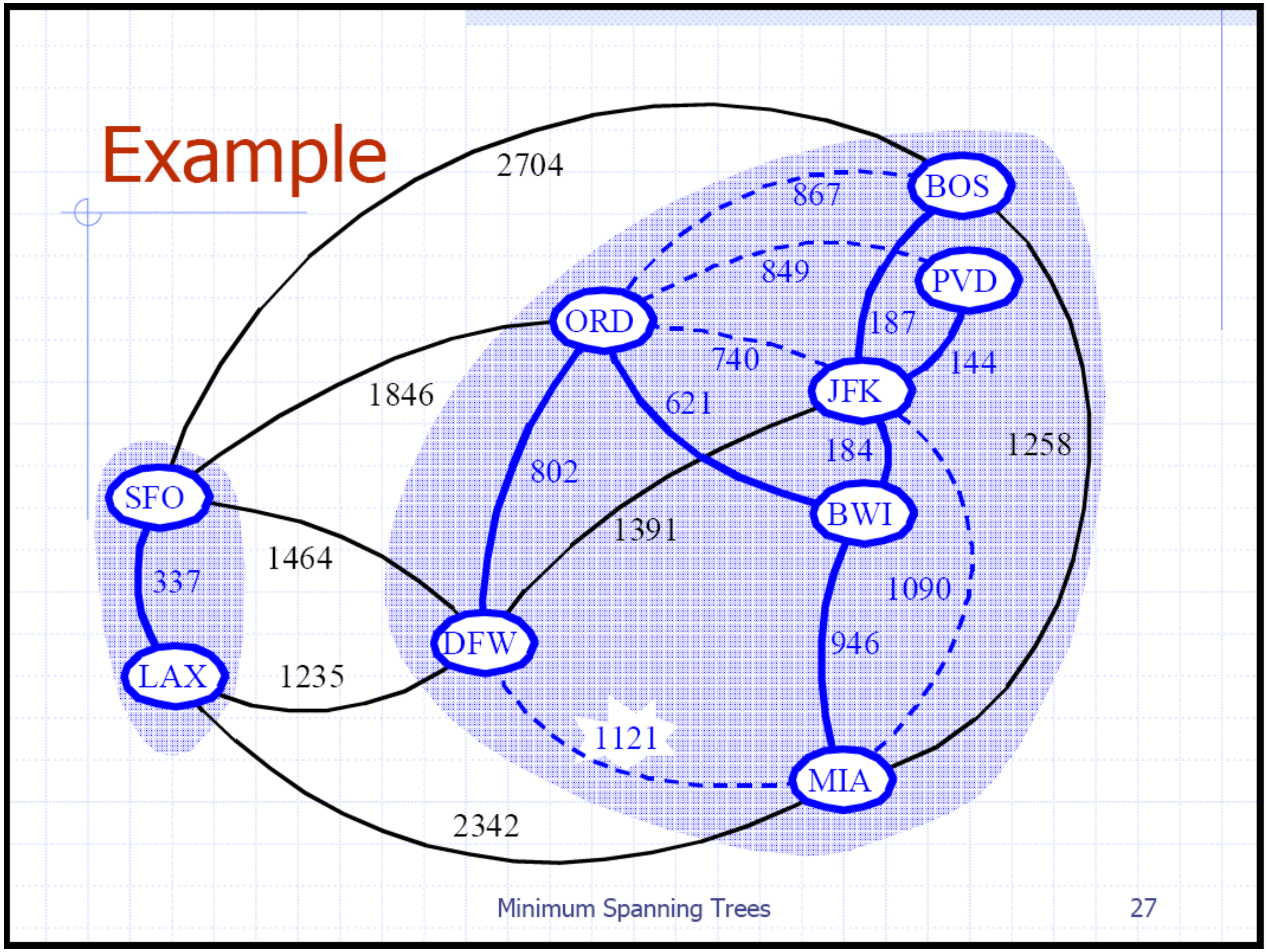


Example

Minimum Spanning Trees

26

Kruskal’s Algorithm Example

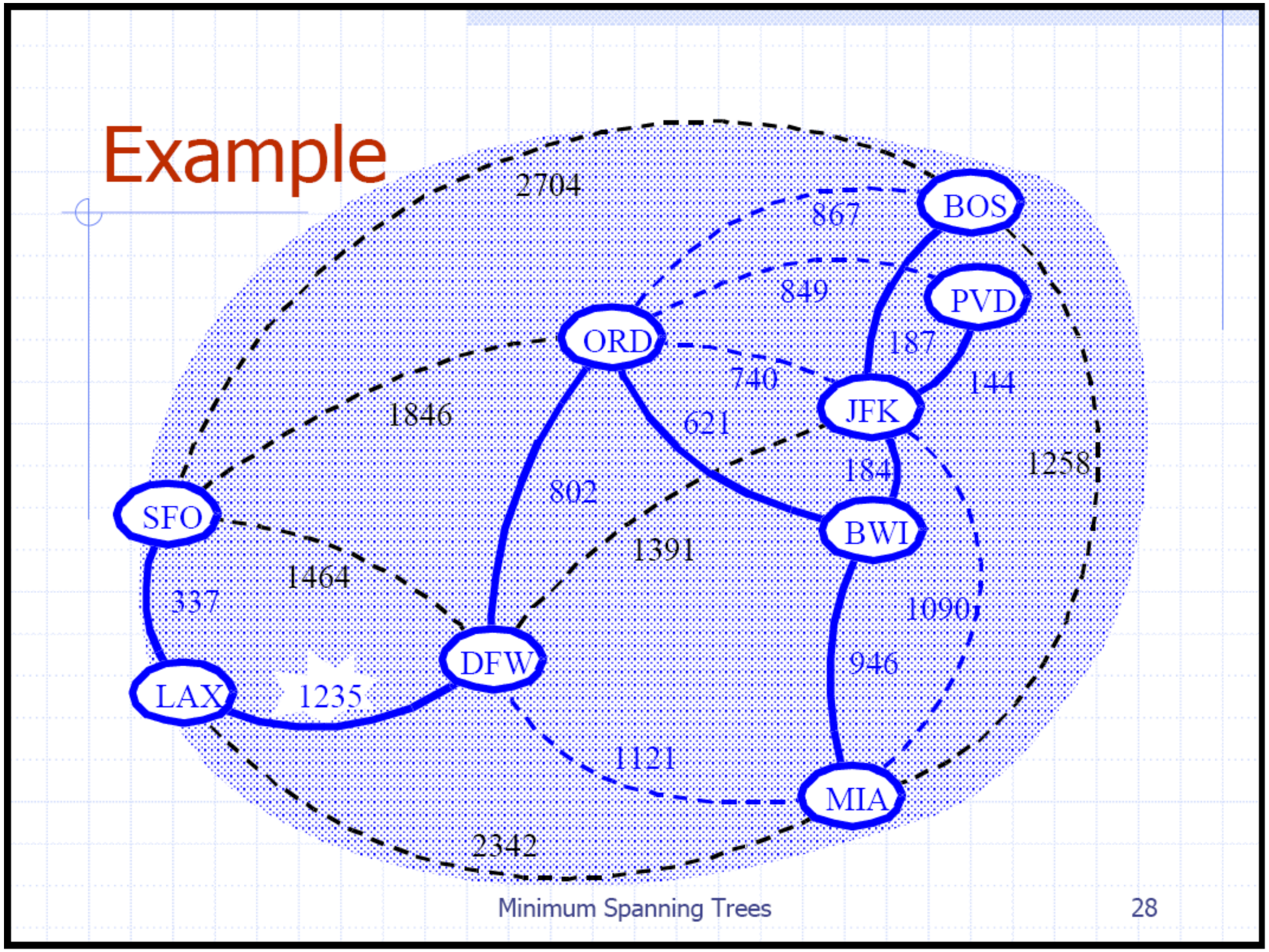


Example

Minimum Spanning Trees

27

Kruskal’s Algorithm Example



Example

Thank You !!!