Minimum Spanning Tree

General Problem: Spanning a Graph

A simple problem: Given a *connected* graph $G=(V,E)$, find a minimal subset of the edges such that the graph is still connected

- A graph $G_2=(V,E2)$ such that G_2 is connected and removing any edge from E_2 makes G_2 disconnected

Observations

1. Any solution to this problem is a tree
   – Recall a tree does not need a root; just means acyclic
   – For any cycle, could remove an edge and still be connected
   – We usually just call the solutions spanning trees

2. Solution not **unique** unless original graph was already a tree

3. Problem ill-defined if original graph not connected
   – We can find a spanning tree per connected component of the graph
   – This is often called a *spanning forest*

4. A tree with **|V|** nodes has **|V|-1** edges
   – This every spanning tree solution has **|V|-1** edges

Motivation

A spanning tree connects all the nodes with as few edges as possible

In most compelling uses, we have a *weighted* undirected graph and want a tree of least total cost

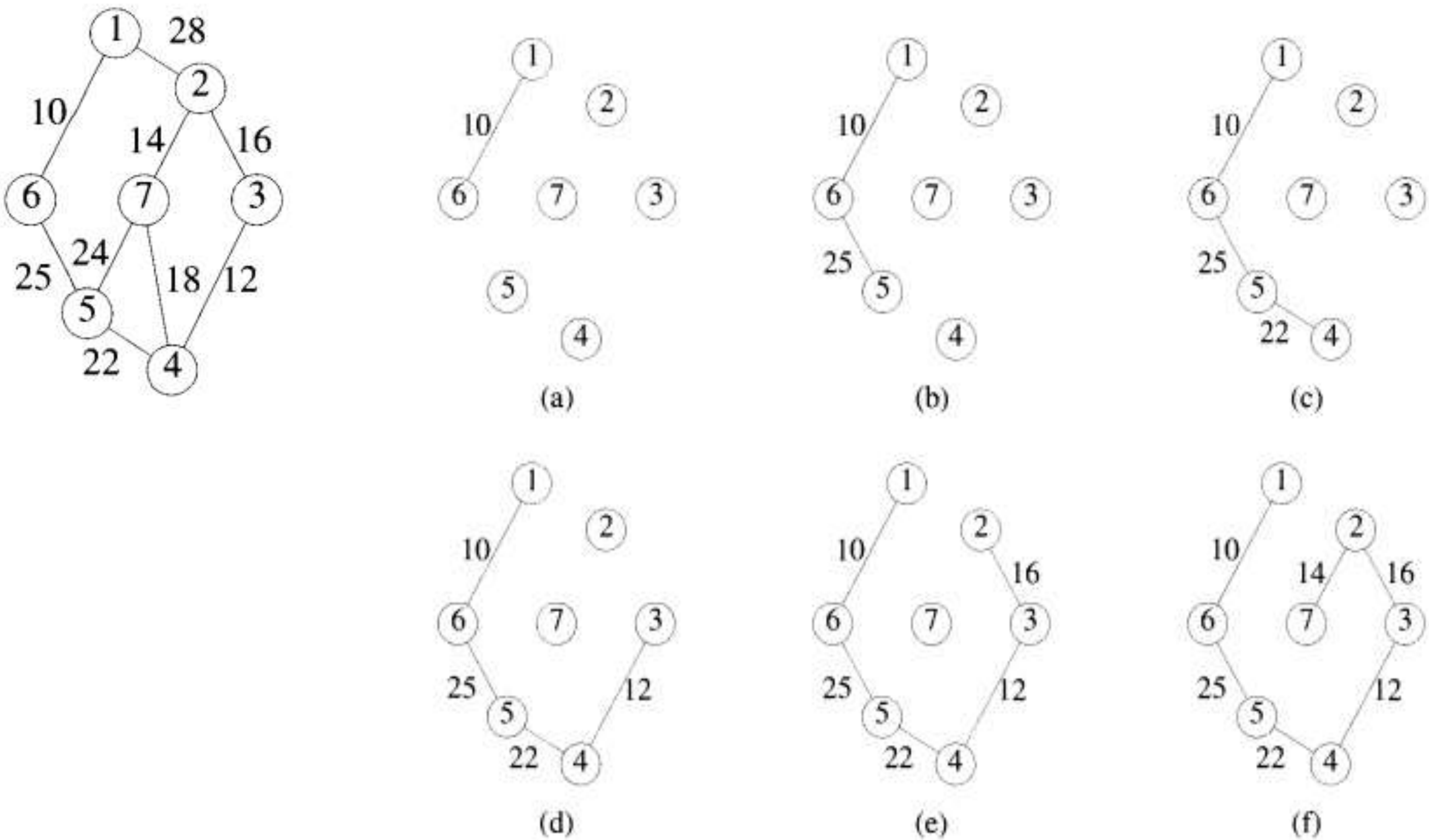
- Minimize electrical wiring for a house or wires on a chip
- Minimize road network

This is the minimum spanning tree problem

Two Approaches to find minimum spanning tree using Greedy

- Prim's Algorithm
- Kruskal's Algorithm

Prim’s Algorithm



Kruskal’s Algorithm