Minimum Spanning Trees
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
Prim’s Algorithm

**Algorithm** Prim (G,s)

**In:** weighted connected graph G and vertex s

**Out:** {compute a minimum spanning tree}

**for** each vertex u of G **do** {
  u.d ← ∞  // distance from vertex s to vertex u
  u.p ← null  // predecessor or parent of vertex u in a shortest paths tree
  u.marked ← false
}

s.d ← 0  // Distance from s to itself is 0

**for** i ← 0 **to** n-1 **do** {
  min ← ∞  // Find unmarked vertex u with minimum distance to s
  **for** each vertex v of G **do**
    if (v.marked = false) and (v.d < min) **then** {
      min ← v.d
      u ← v
    }
  u.marked ← true  // Relax all edges incident on vertex u
  **for** each edge (u,v) incident on u **do**
    if length(u,v) < v.d **then** {
      v.d ← length(u,v)
      v.p ← u
    }
}
Example
Example (contd.)
Analysis of Prim’s Algorithm

Using an adjacency matrix:

\[ f(n,m) = c_1n + \sum_{u \in G}(c_4+c_2n+c_3n) = c_1n + c_4n + c_2n^2 + c_3n^2 \text{ is } O(n^2) \]

Using an adjacency list:

\[ f(n,m) = c_1n + \sum_{u \in G}(c_4+c_2n+c_3\text{deg}(u)) = c_1n + c_4n + c_2n^2 + 2c_3m \text{ is } O(n^2) \]