ZAMAN UNIVERSITY

Data Structures and Algorithms

Chapter 6

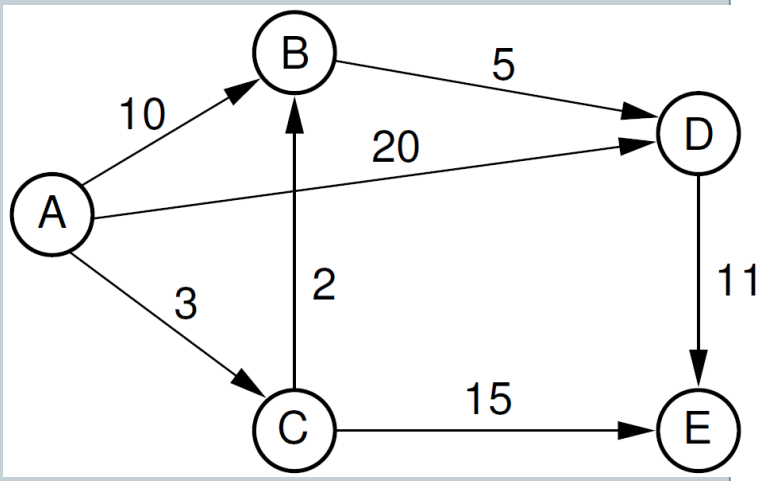
Graphs

Outline

- Terminology and Representations
- Graph Traversals
- Shortest-Paths Problems
- Minimum-Cost Spanning Trees

Shortest-Paths Problems

- On a road map, a road connecting two towns is typically labeled with its distance (in number)
- In graph, the number represents the distance between two vertices
- These numbers may be called **weights**, **costs**, or **distances**, depending on the application
- Thus, a graph, a typical problem is to find the total length of the shortest path between two specified vertices.



Outline

- Terminology and Representations
- Graph Traversals
- Shortest-Paths Problems
  - Single-Source Shortest Path
- Minimum-Cost Spanning Trees

Single-Source Shortest Paths

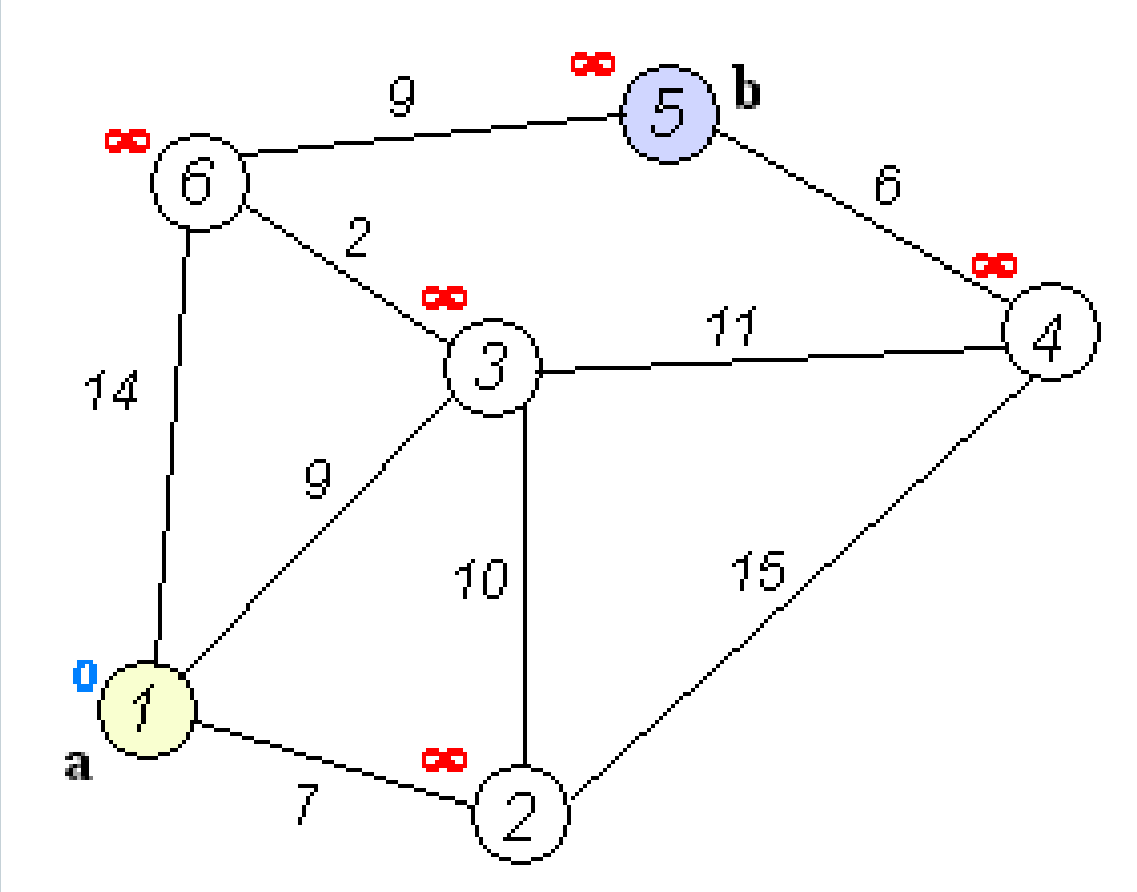
- Single-Source Shortest Paths – sometimes called **Dijkstra** algorithm
- The algorithm is to find shortest paths from *s* (starting vertex) to every other vertex in *G*.
- It is the worst case, while finding the shortest path from *s* to *t*, we might find the shortest paths from *s* to every other vertex as well.

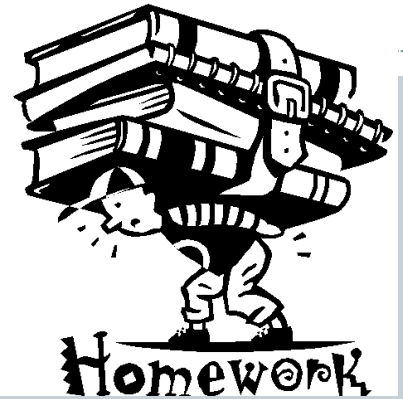
- In computer networks to broadcast a message from a computer to all other computers, the goal is to find the shortest paths from source computer to other.

Dijkstra Algorithm Pseudo-Code


```
1. function Dijkstra(Graph, s):
2.   Create Unvisited List Q = {}

3.   for each vertex v in Graph:   // Initialization
4.       dist[ v ] ← INFINITY      // Unknown distance from source to v
5.       prev[ v ] ← UNDEFINED     // Previous node in optimal path from source
6.       add v to Q                // All nodes initially in Q (unvisited nodes)

7.   dist[ s ] ← 0                 // Distance from source to source

8.   while Q is NOT empty:
9.     u ← vertex in Q with min dist[u] // Source node will be selected first
10.    remove u from Q

11.    for each neighbor v of u:          // where v is still in Q.
12.      alt ← dist[u] + length(u, v)

13.      if alt < dist[v]:                // A shorter path to v has been found
14.           dist[v] ← alt
15.           prev[v] ← u
```

Dijkstra Algorithm Animation





Write functions of Dijkstra (Single-Source Shortest Paths) base on pseudo-code in the slide.

To be continued…