Project 5

Soumya Basu

Department of Computer Science
Cornell University

September 18, 2015

Administrivia

- Project 5 is due on November 18

- Will be implementing a Link State routing protocol

The Five Parts

- Connecting two nodes
- Dijkstra's algorithm
- Flooding
- Link State Routing
- (Optional) Application!

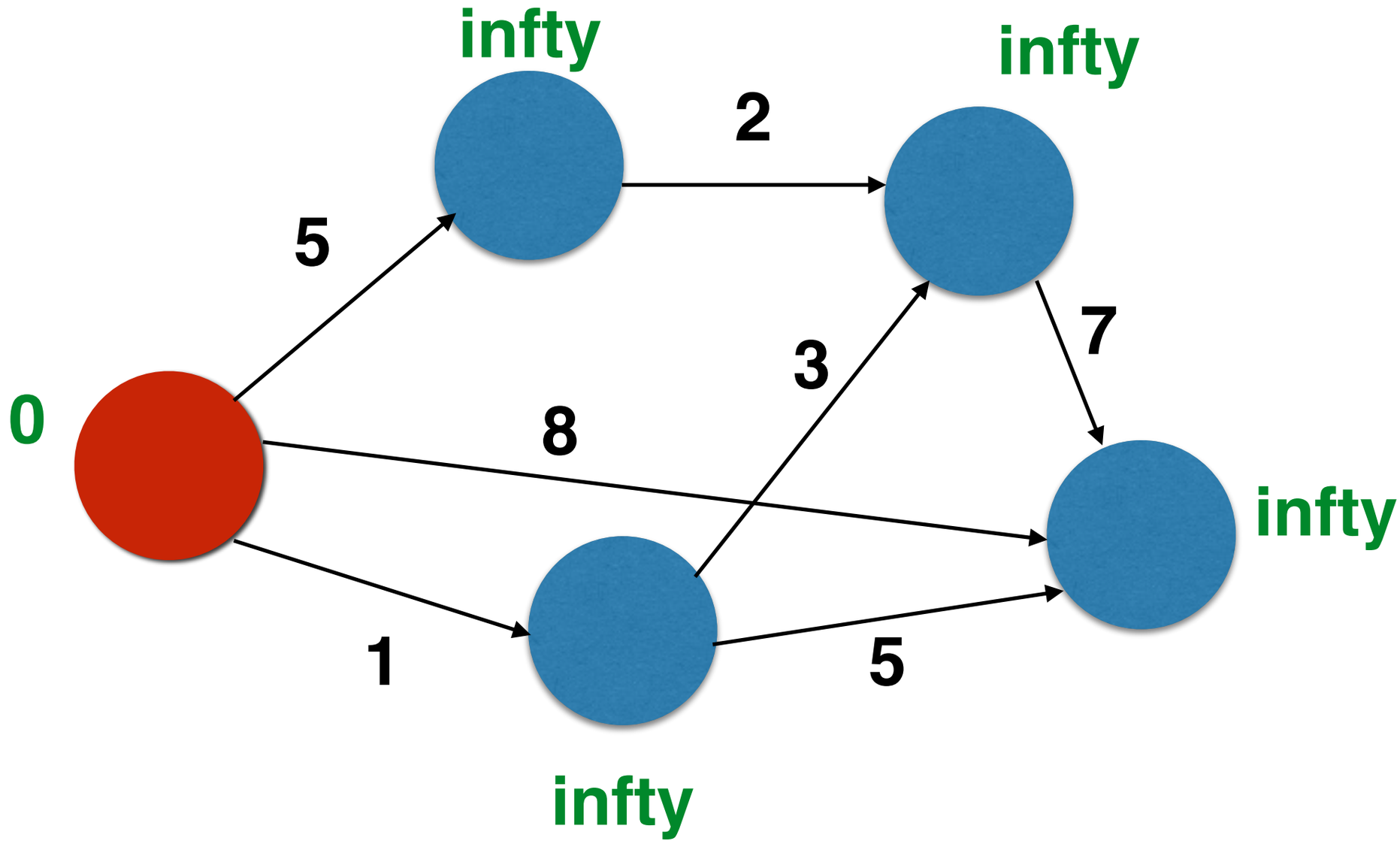
Part 0

- Read the skeleton code
- You will need to interact with most of it

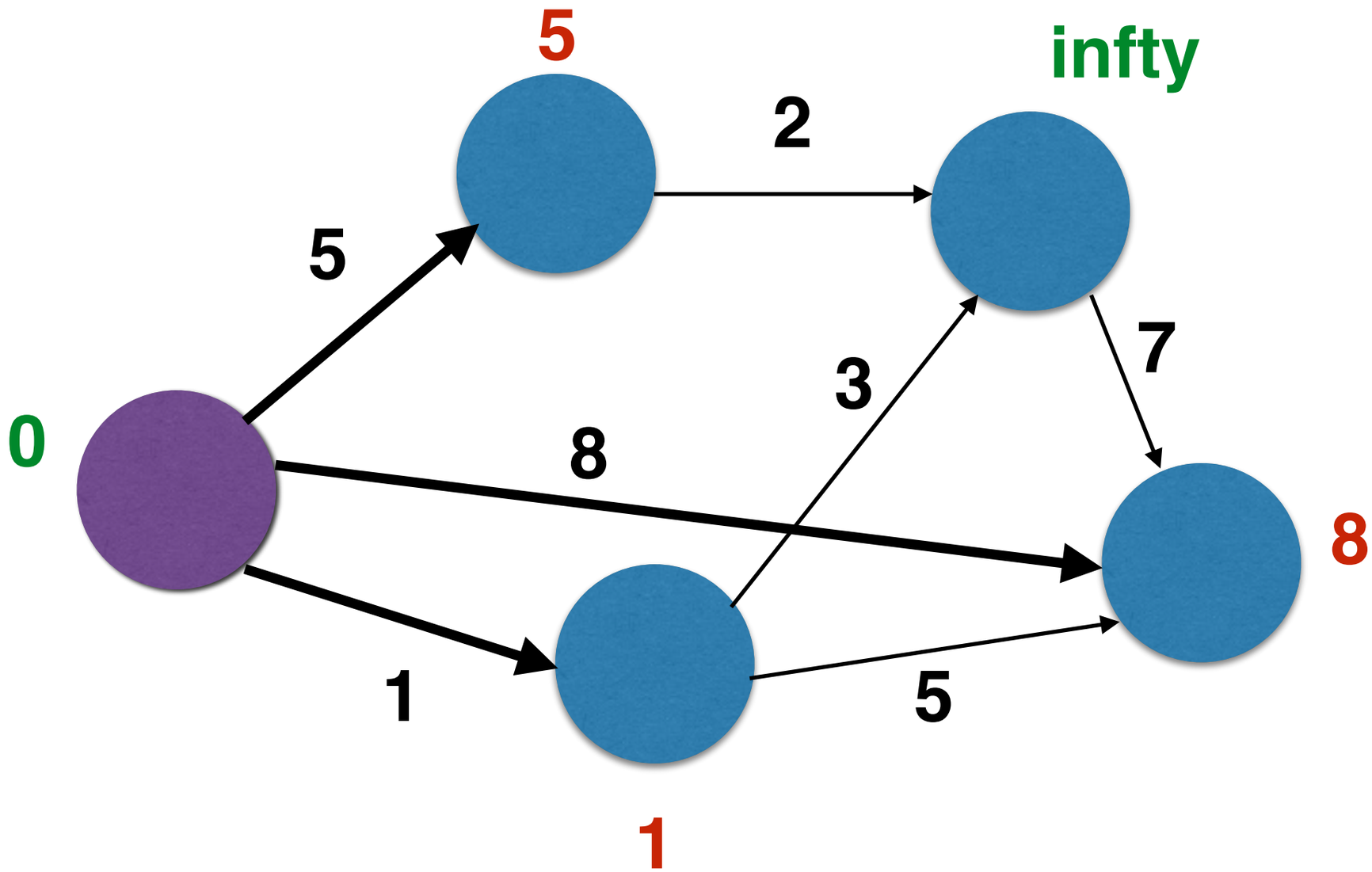
Connecting two nodes

- Need to read up on the socket API calls
- Important functions:
  - socket, fcntl, setsockopt, bind
  - Use man pages to see the options
- Not many lines of code
- Format: C<IPaddr:port>

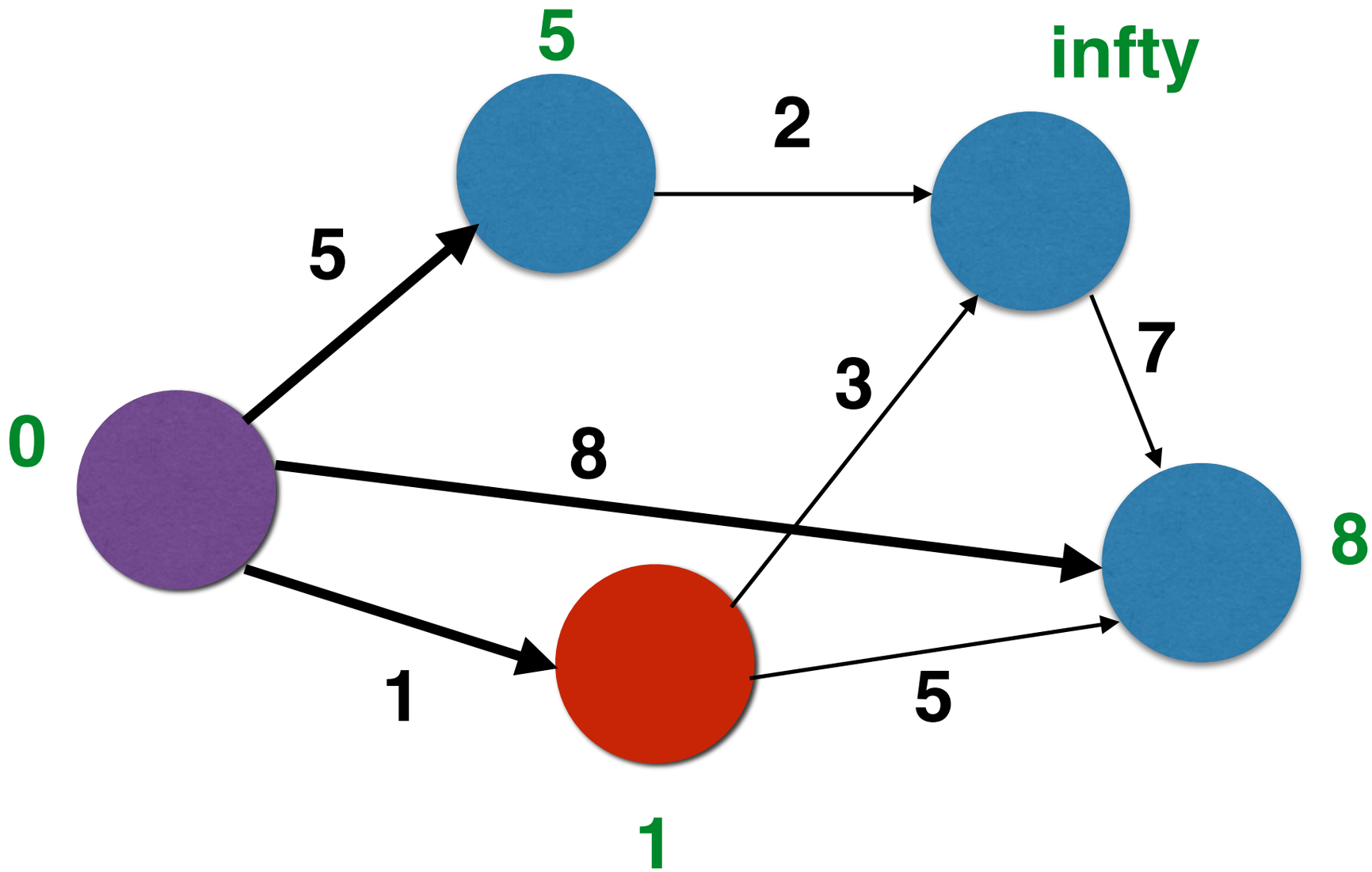
Dijkstra's Algorithm



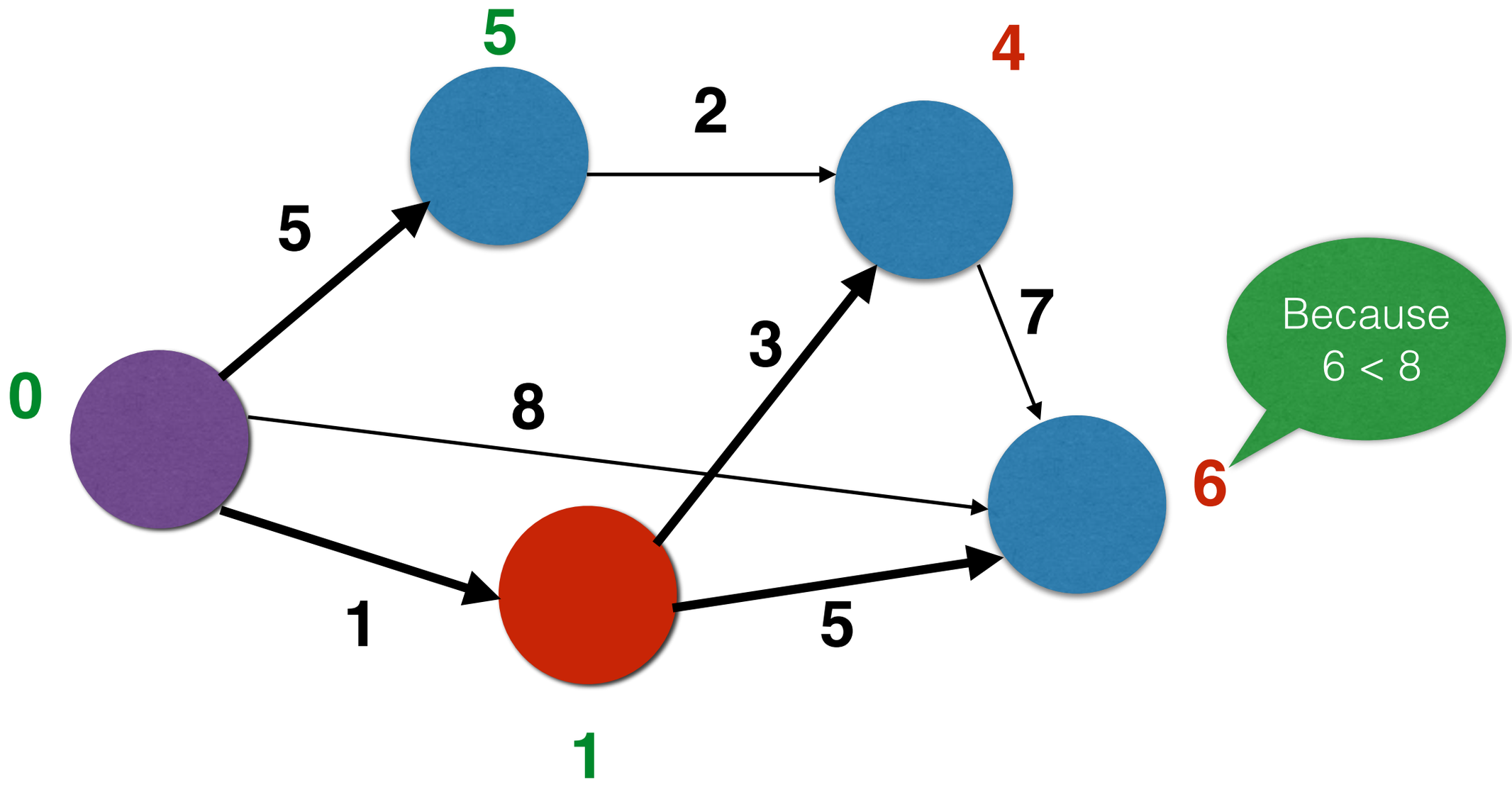
Dijkstra's Algorithm



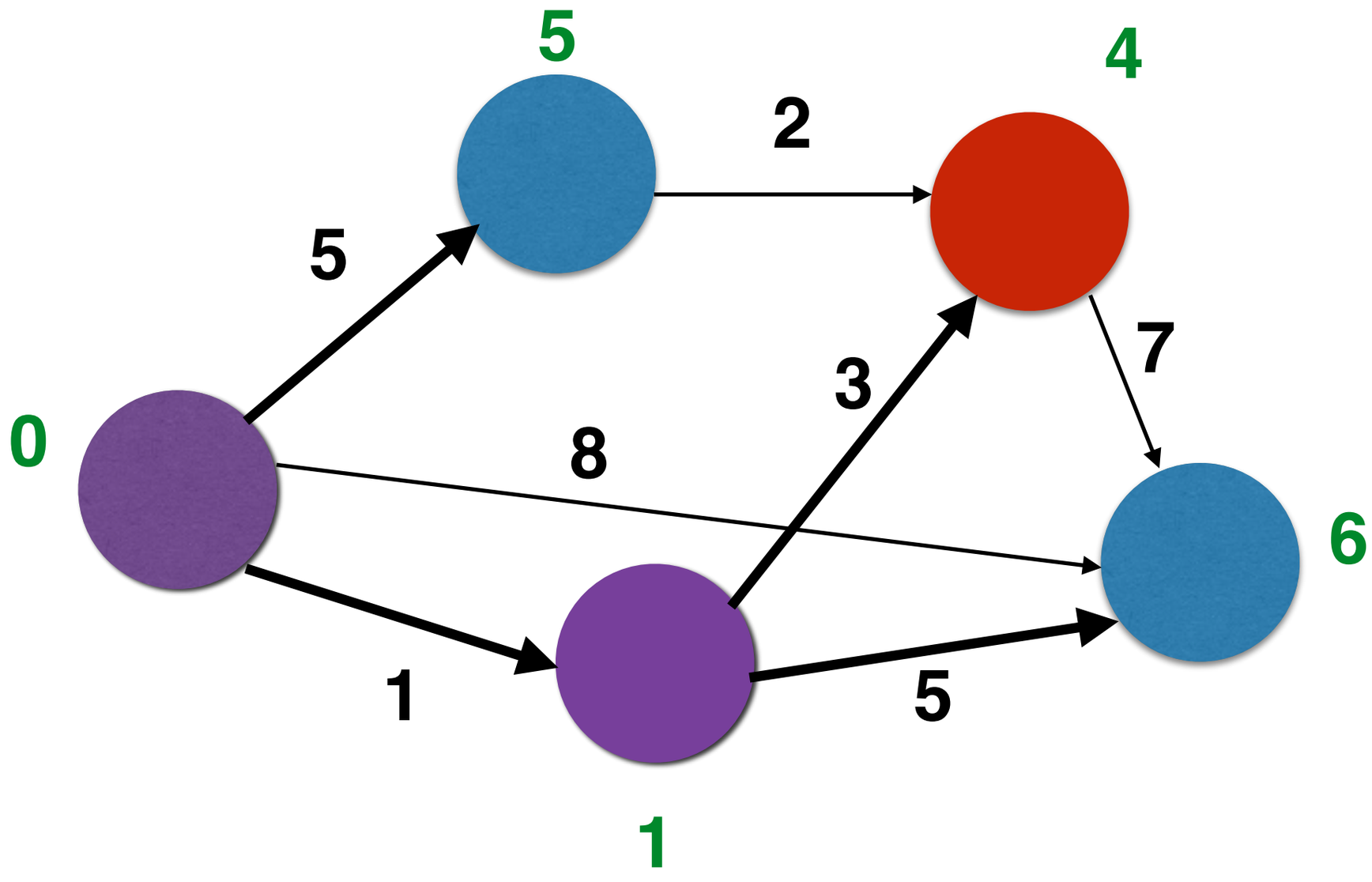
Dijkstra's Algorithm



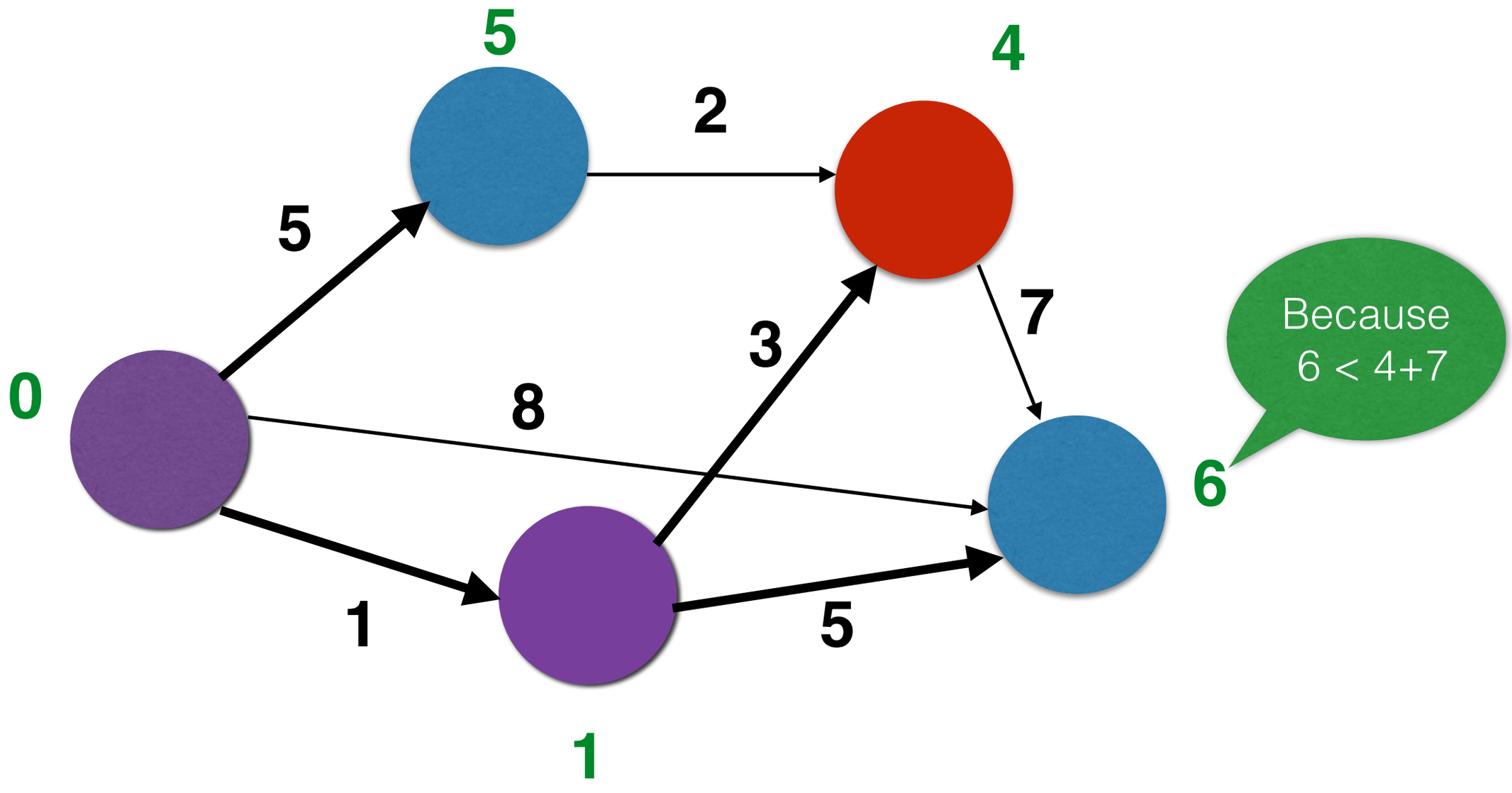
Dijkstra's Algorithm



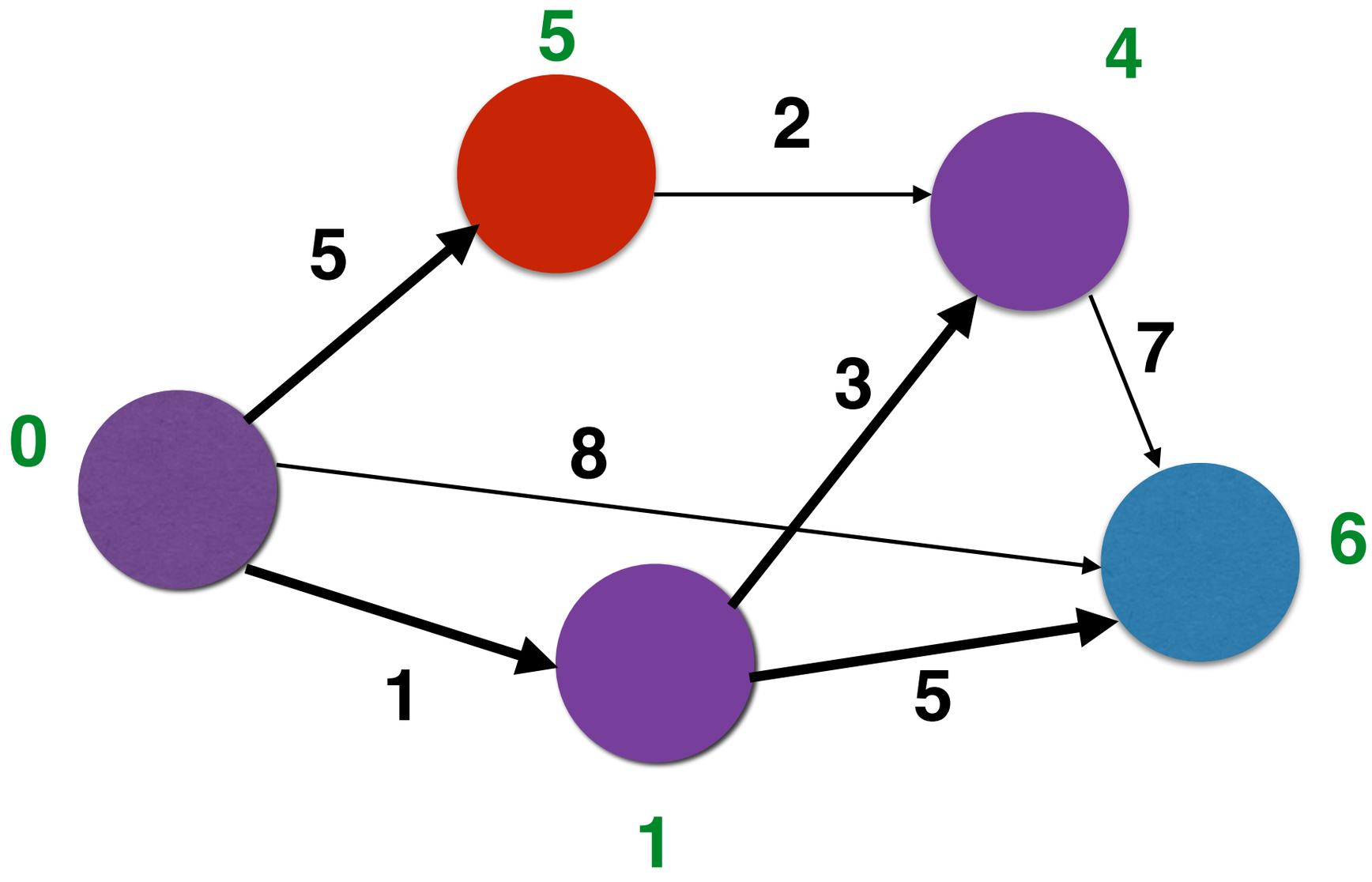
Dijkstra's Algorithm



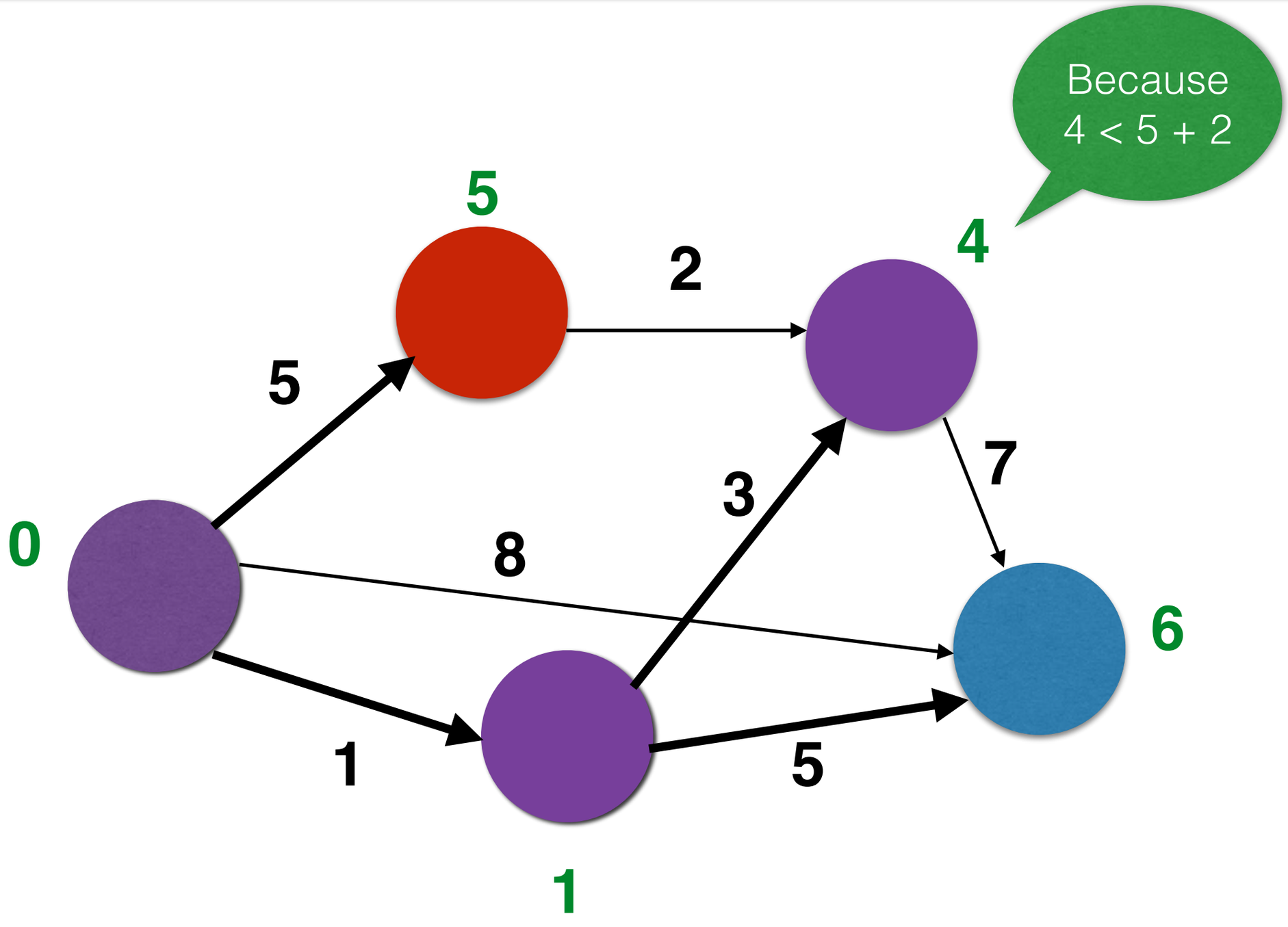
Dijkstra's Algorithm



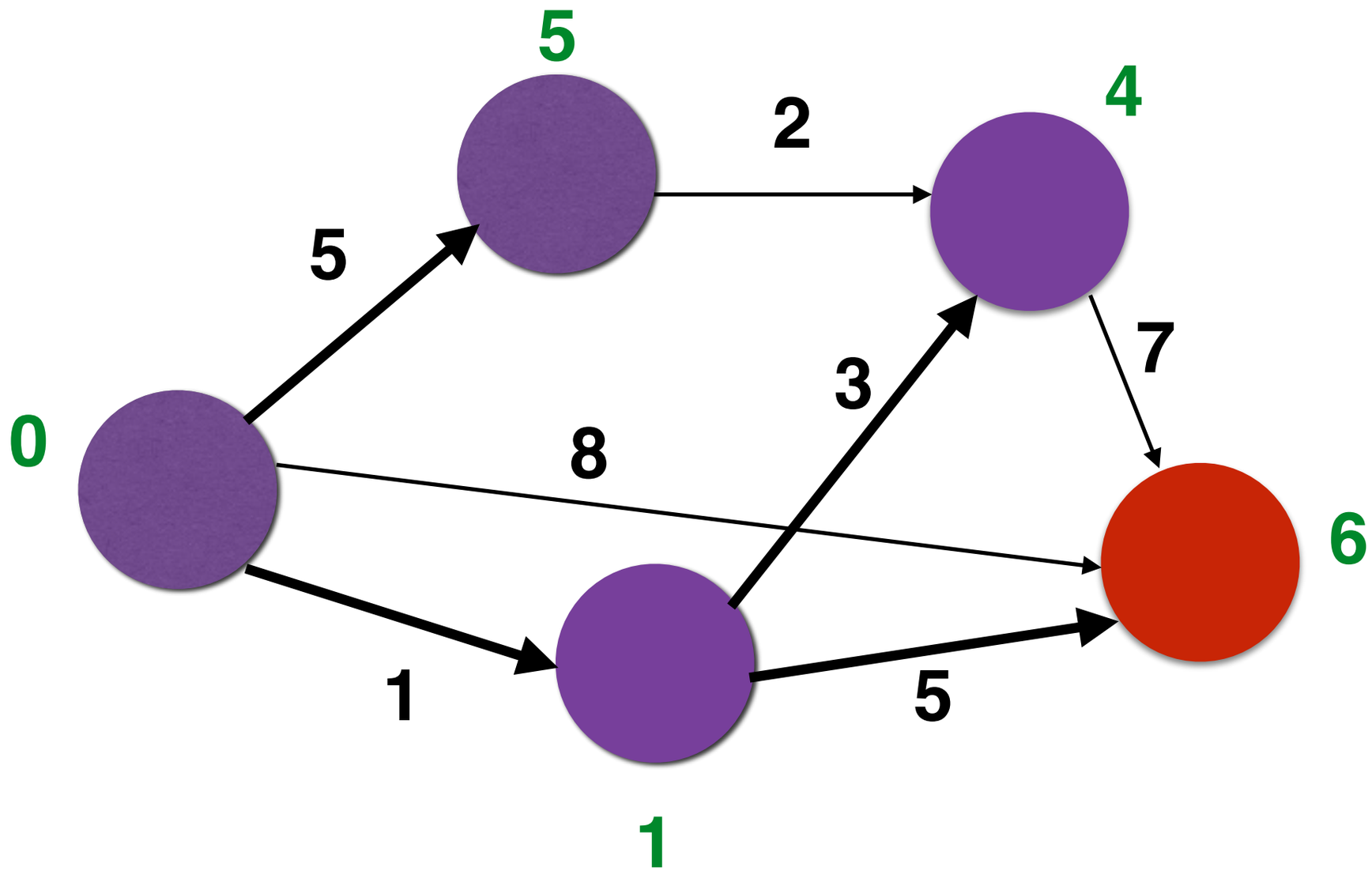
Dijkstra's Algorithm



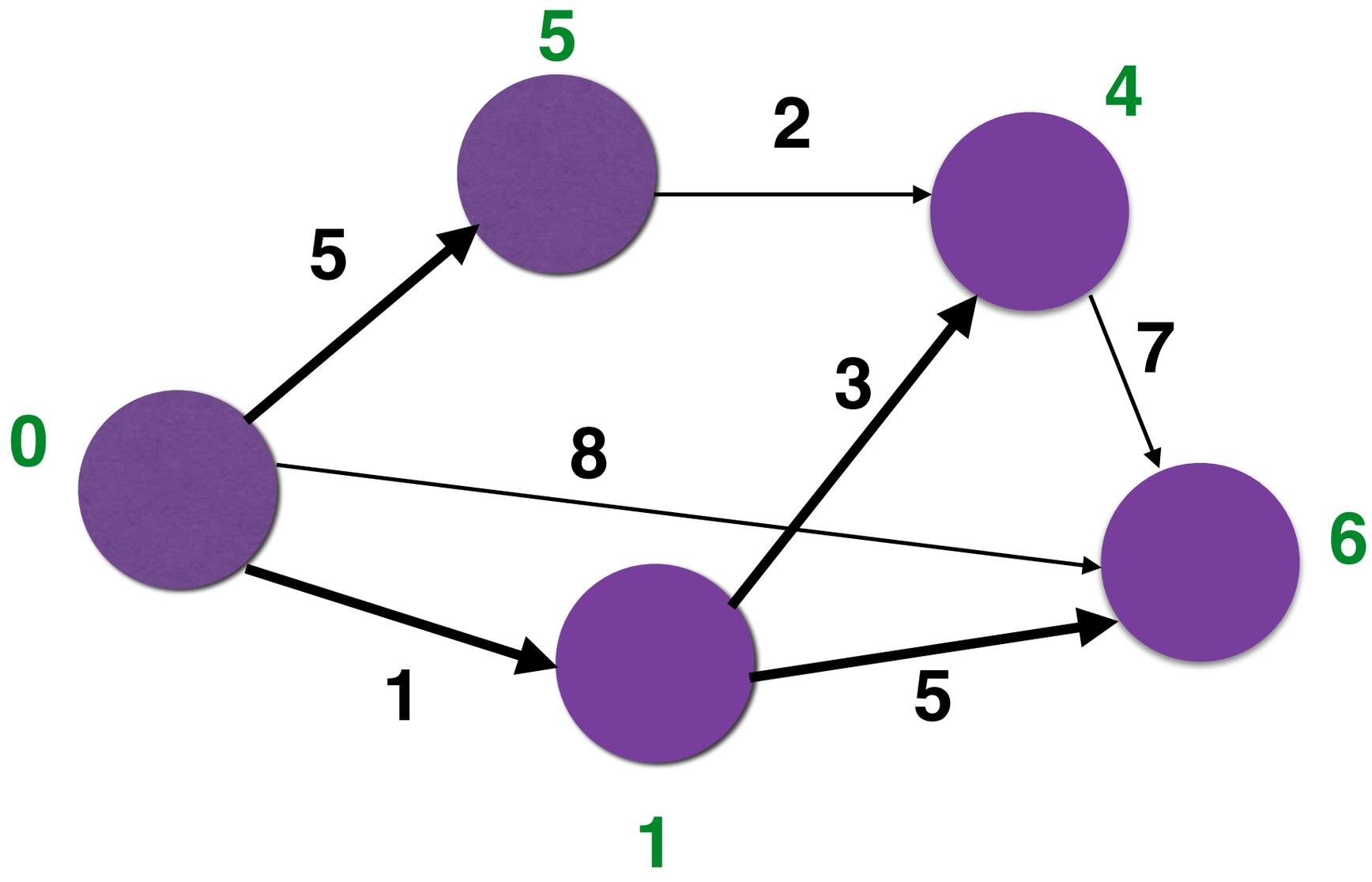
Dijkstra's Algorithm



Dijkstra's Algorithm



Dijkstra's Algorithm

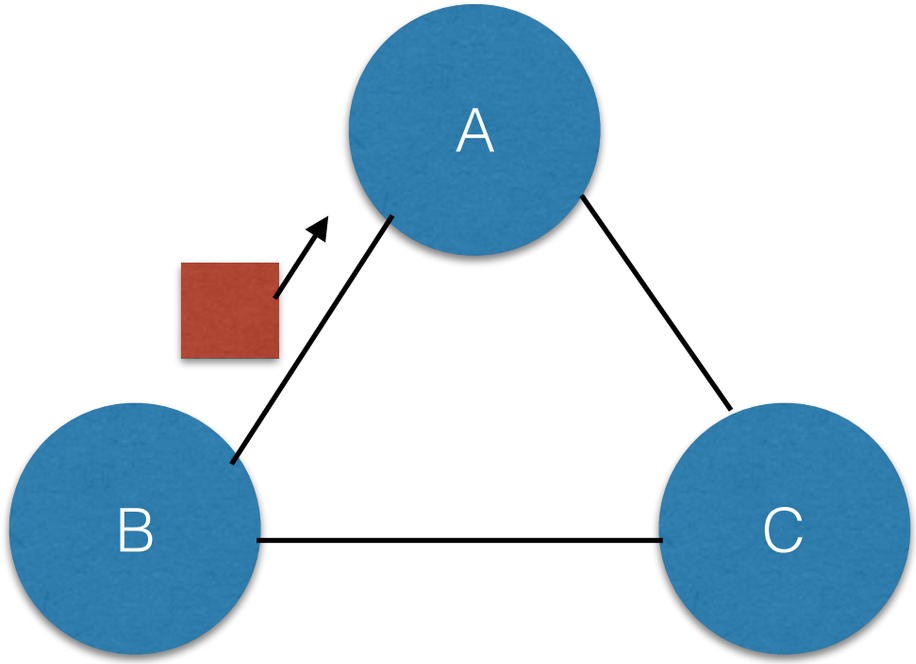


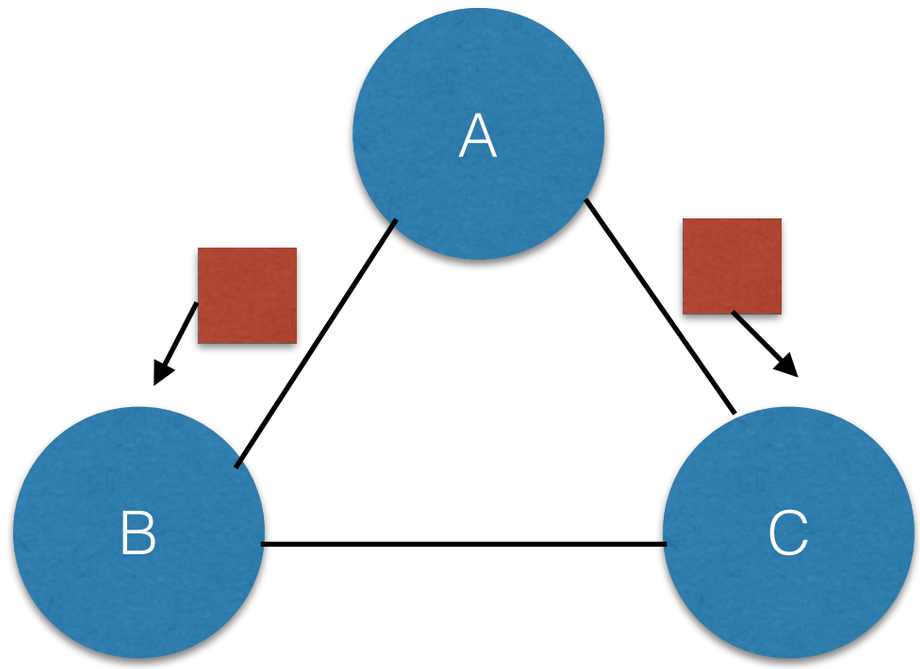
Flooding

- How do you figure out the link state?
- Broadcast!
- You only have point-to-point though…
- Gossip protocol
  - G<srcIP:srcPort>/counter/payload\n
  - Payload will change

Flooding

- Naive implementation:
  - When you hear a message, tell EVERYONE!

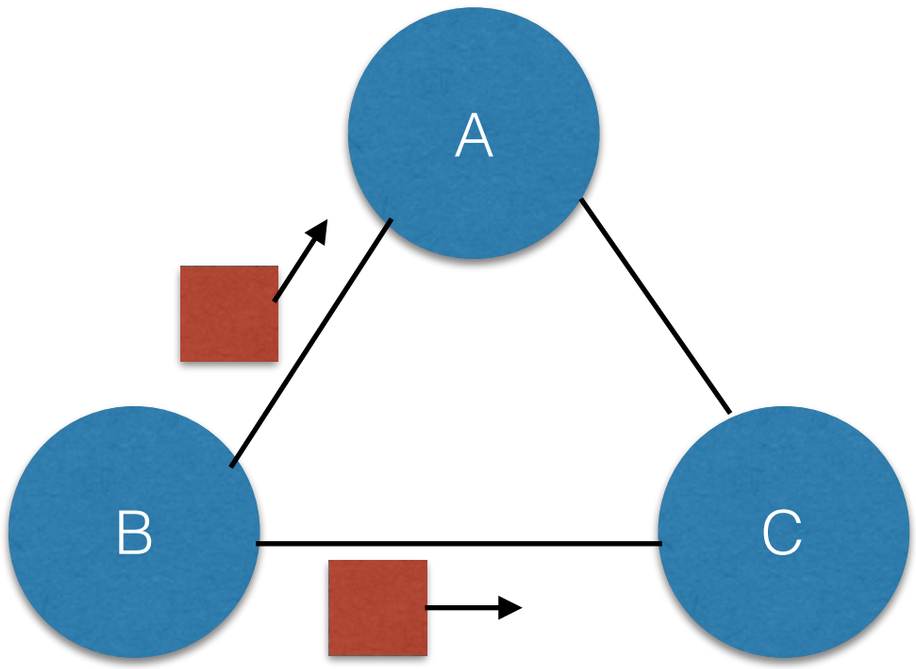




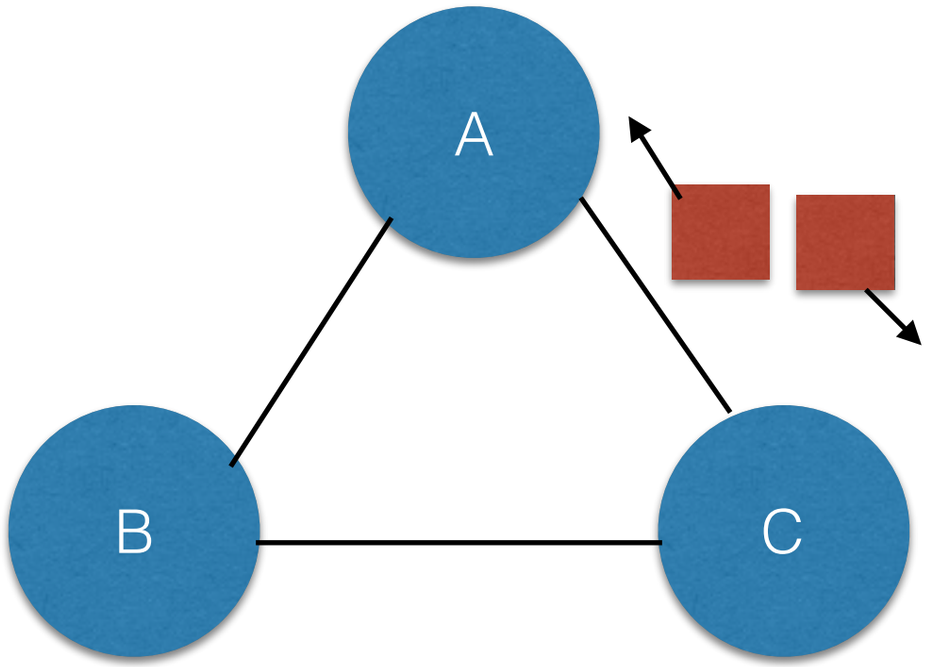
Flooding

- Broadcast storms are not good…
- We make two optimizations:
  - Don’t tell the person that sent you the message
  - Don’t spread around messages you’ve seen already

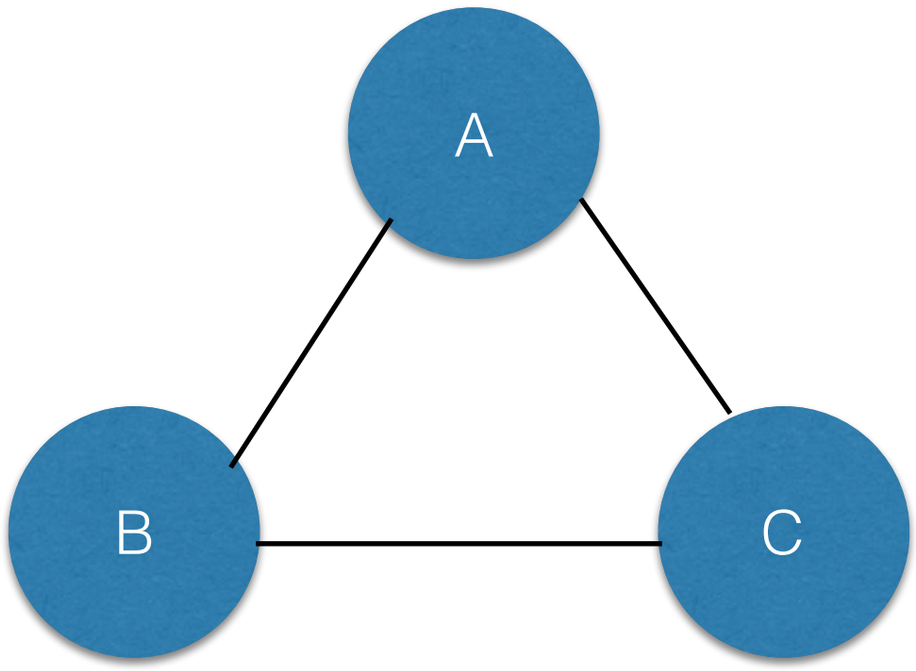
Flooding



Flooding



Flooding



Flooding

- Can we do even better?
  - Yes, but this is good enough!
- What to flood?
  - Your connection state every time it changes
  - A new connection, a dead connection, etc.
  - Gossip payload: “;<addr1:port1>;<addr2:port2>…”

Link State Routing

- You have most of the pieces already!
- When you see a message, send it along the Dijkstra tree
- Most of the work here is the send message implementation
- Send format: S<dstIP:dstPort>/TTL/payload\n
- Make sure to decrement TTL

Optional Applications

- Mapping IP addresses to names (DNS)
- Anything else you want!