

# **15.082J & 6.855J & ESD.78J Visualizations**

---

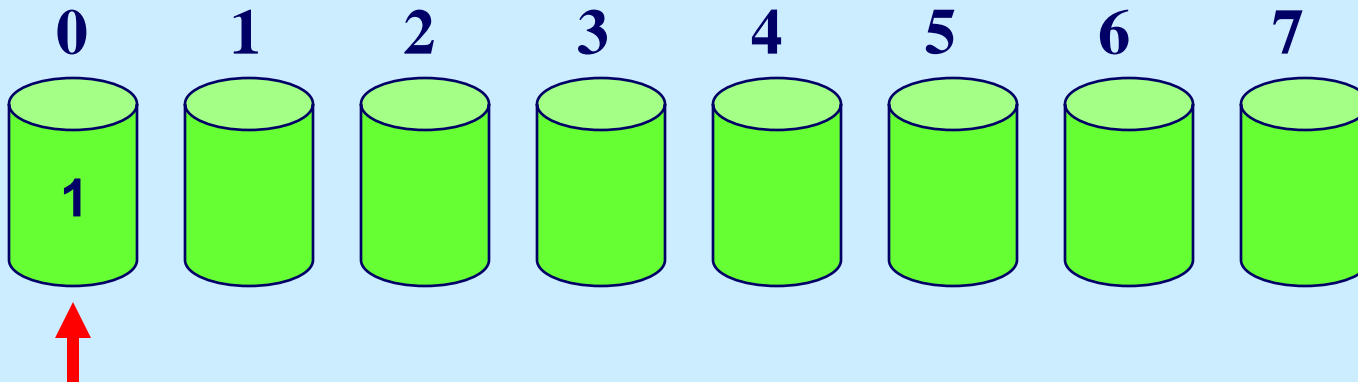
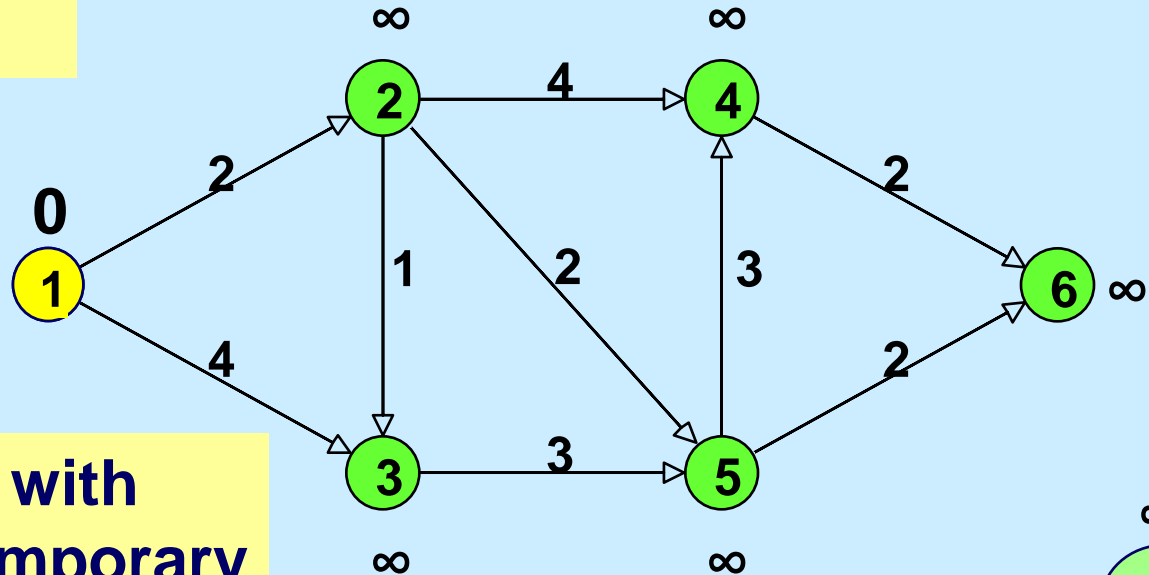
**Dijkstra's Algorithm with simple buckets  
(also known as Dial's algorithm)**

# An Example

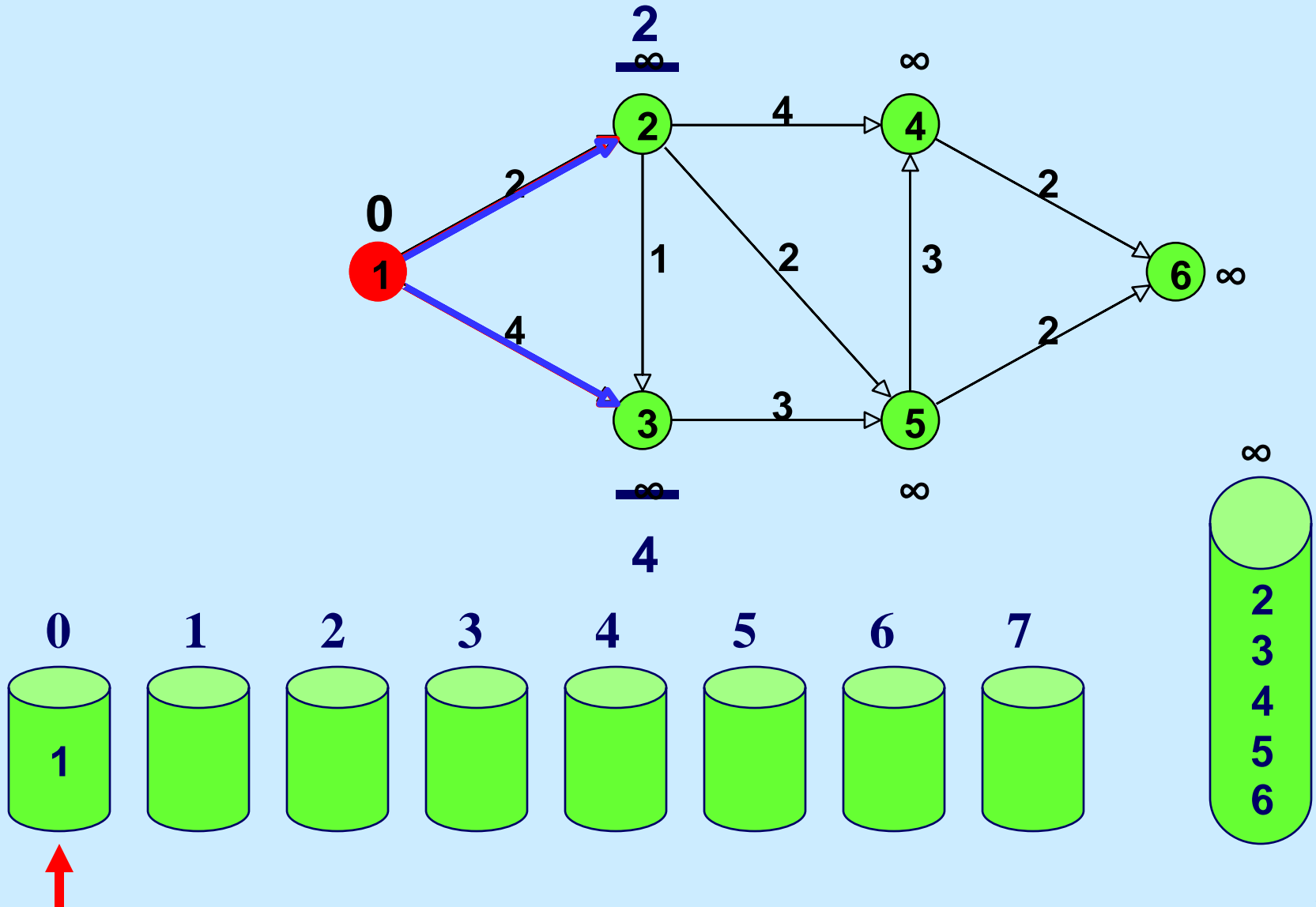
Initialize distance labels

Initialize buckets.

Select the node with the minimum temporary distance label.

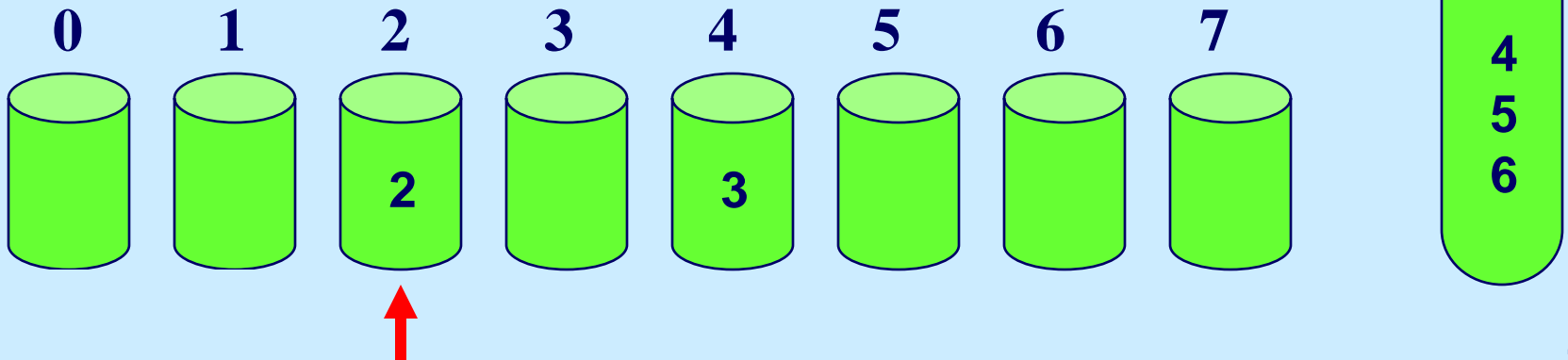
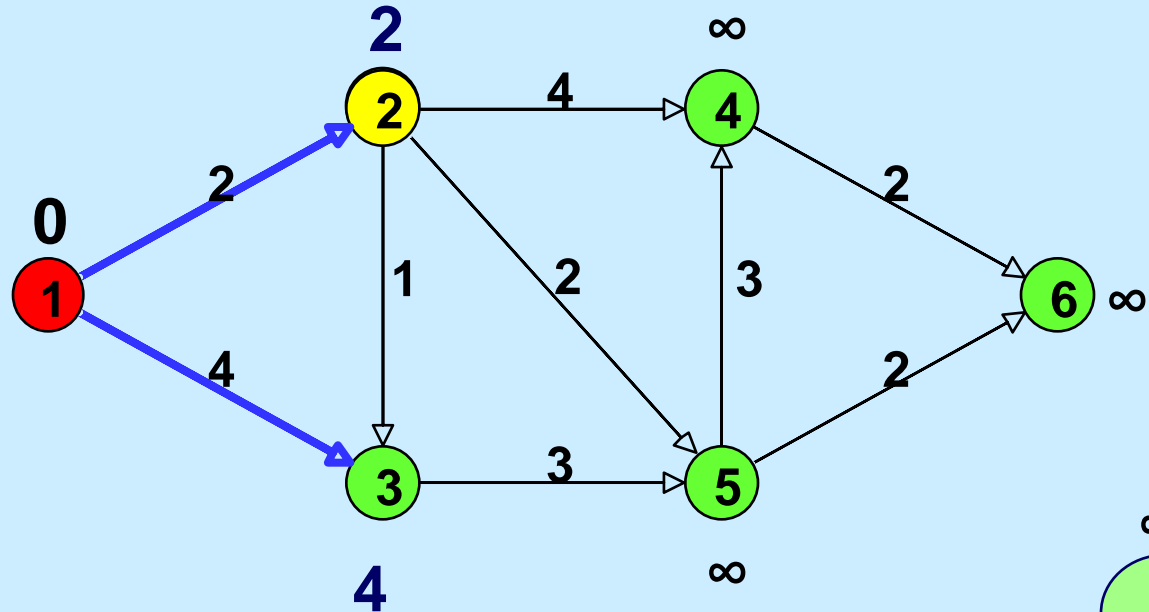


# Update Step

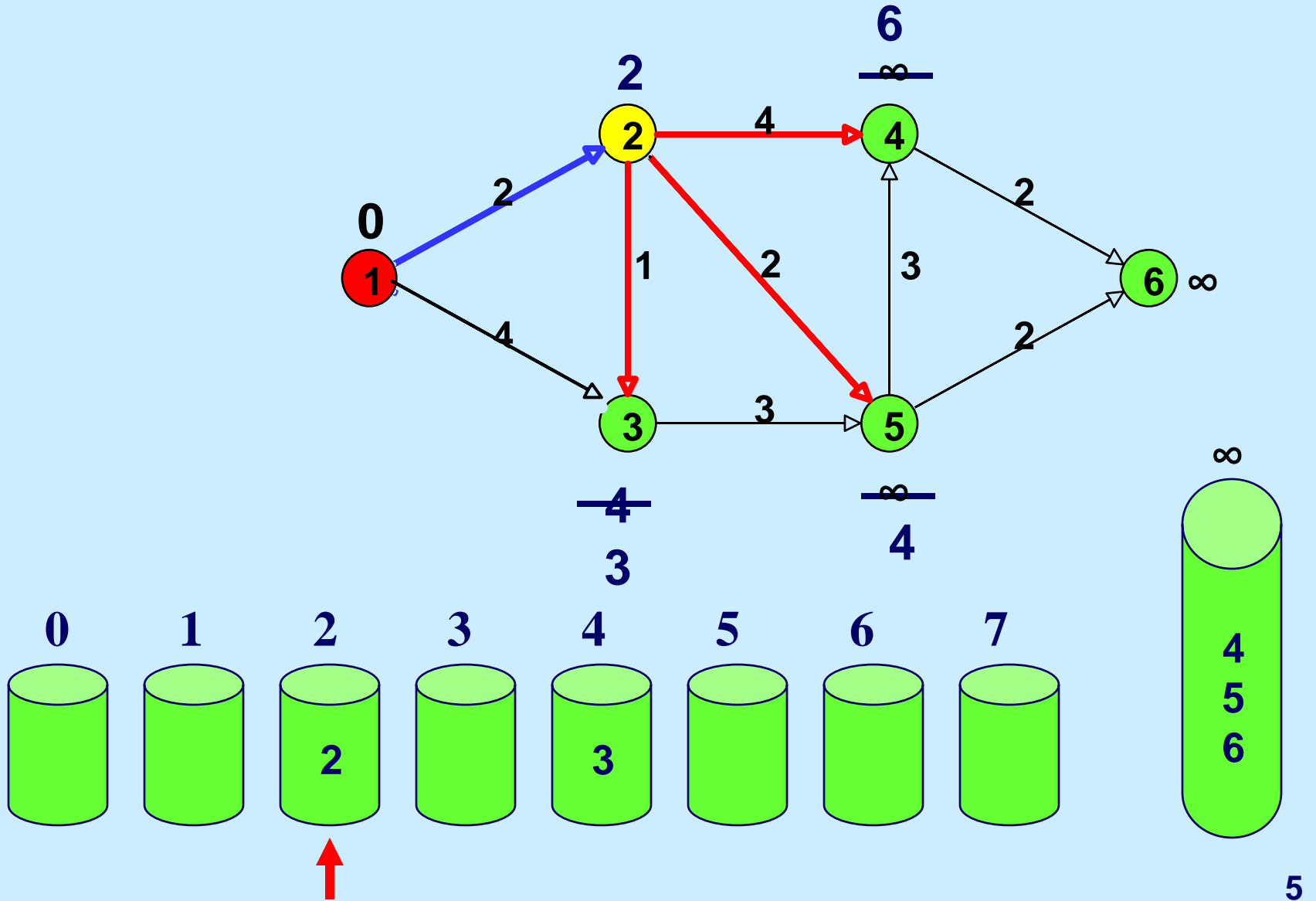


# Choose Minimum Temporary Label

Find Min by starting at the leftmost bucket and scanning right till there is a non-empty bucket.

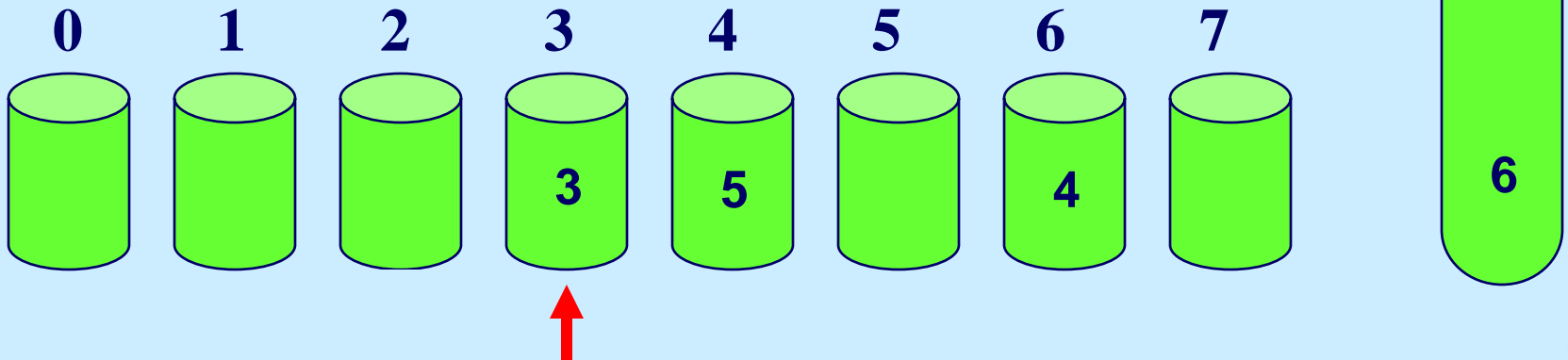
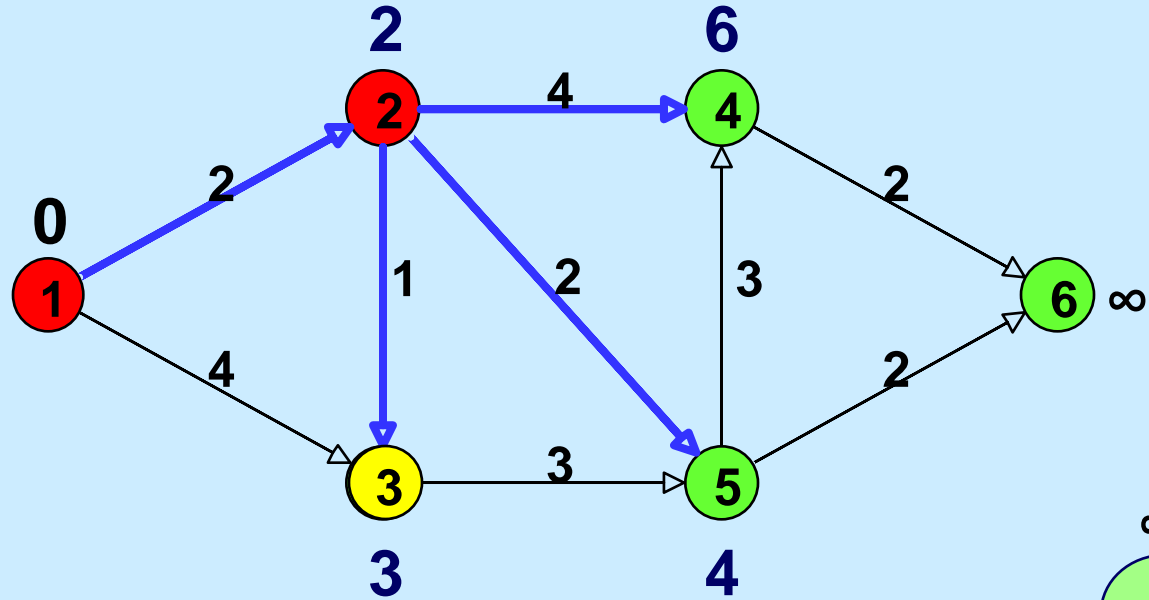


# Update Step

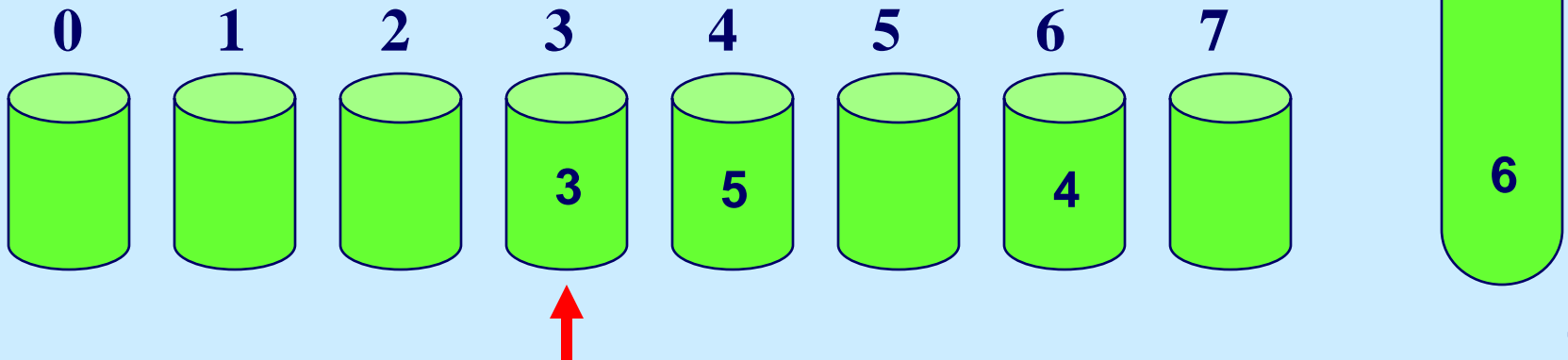
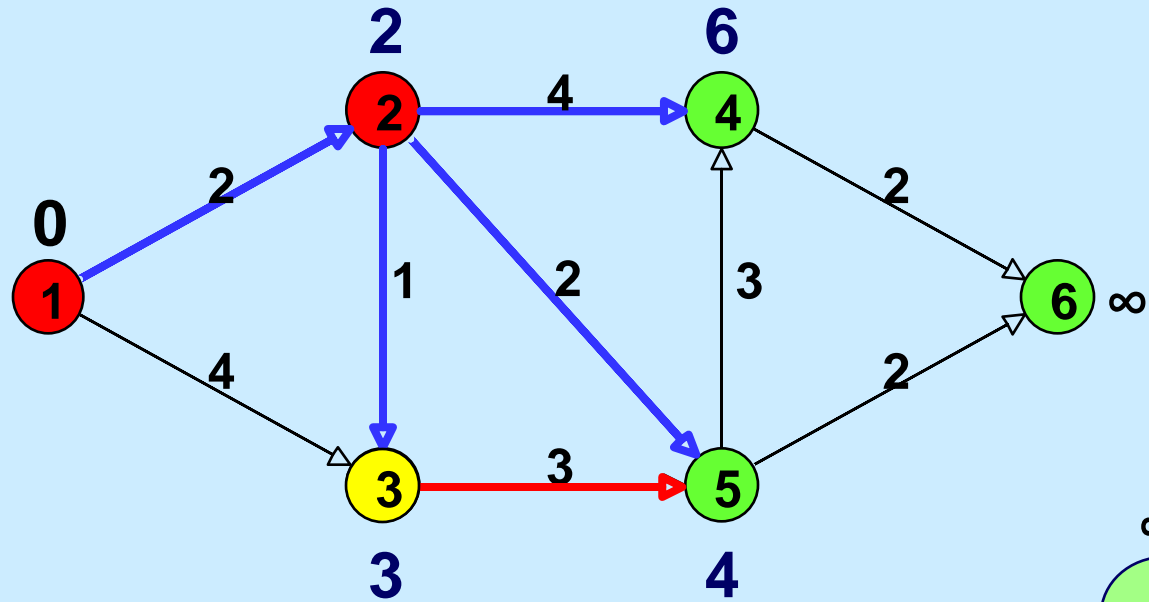


# Choose Minimum Temporary Label

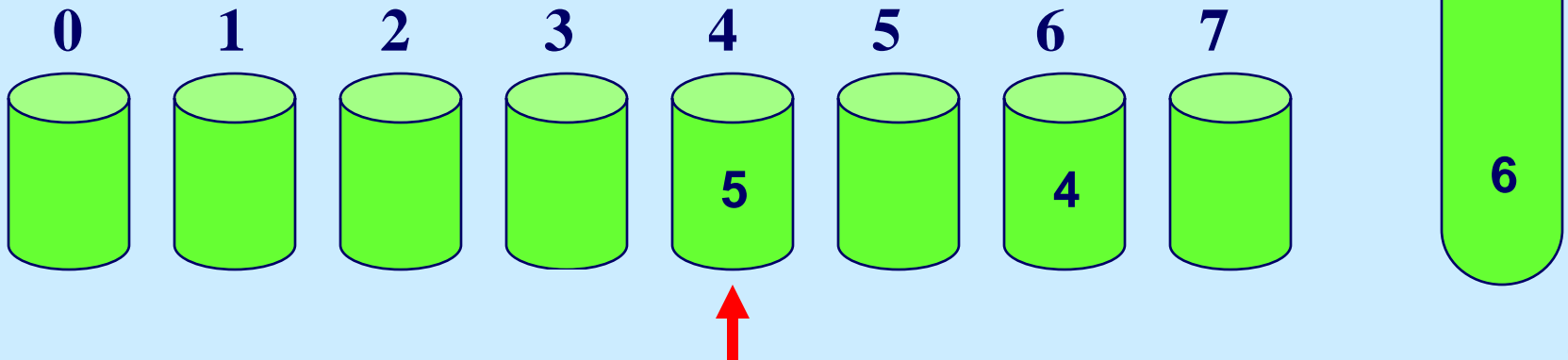
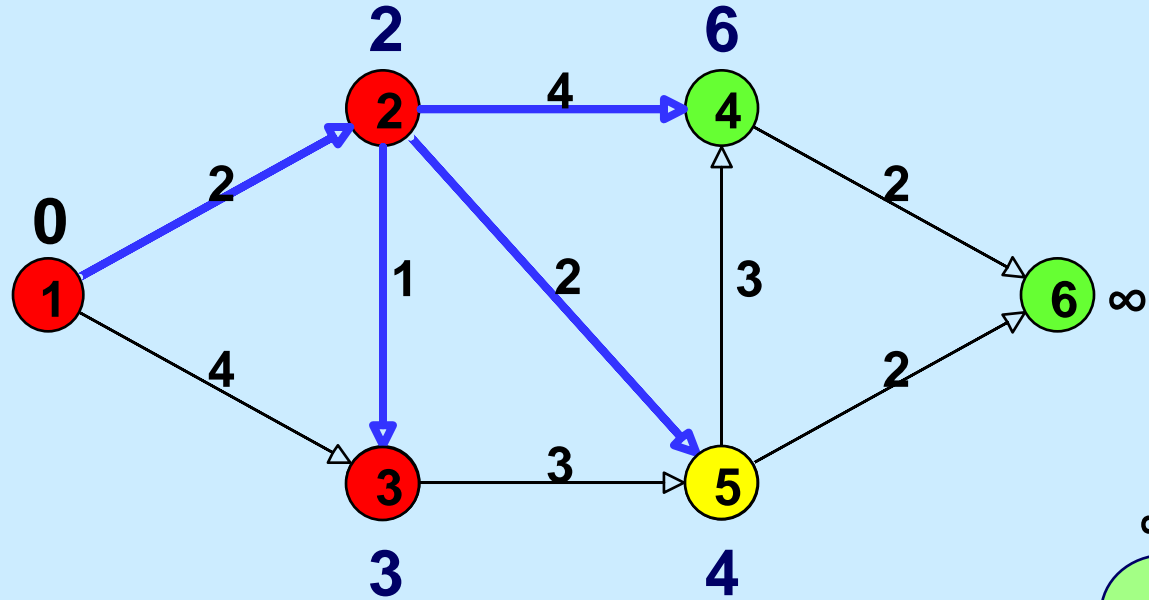
Find Min by starting at the leftmost bucket and scanning right till there is a non-empty bucket.



# Update

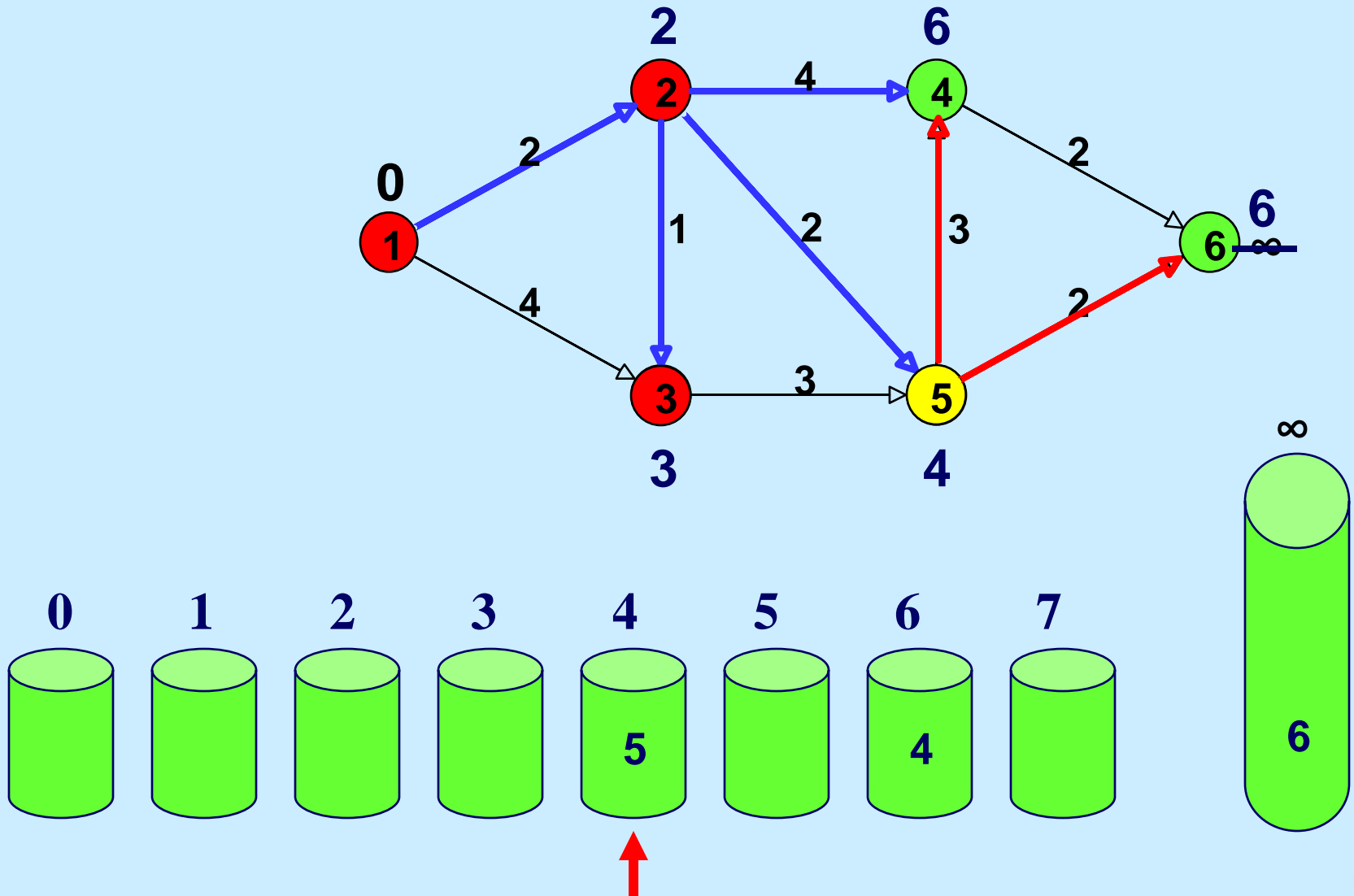


# Choose Minimum Temporary Label

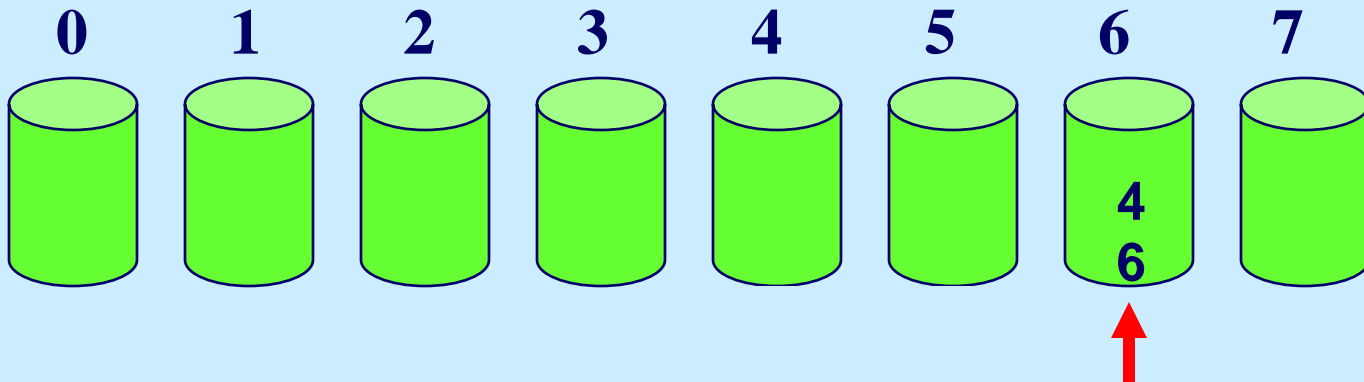
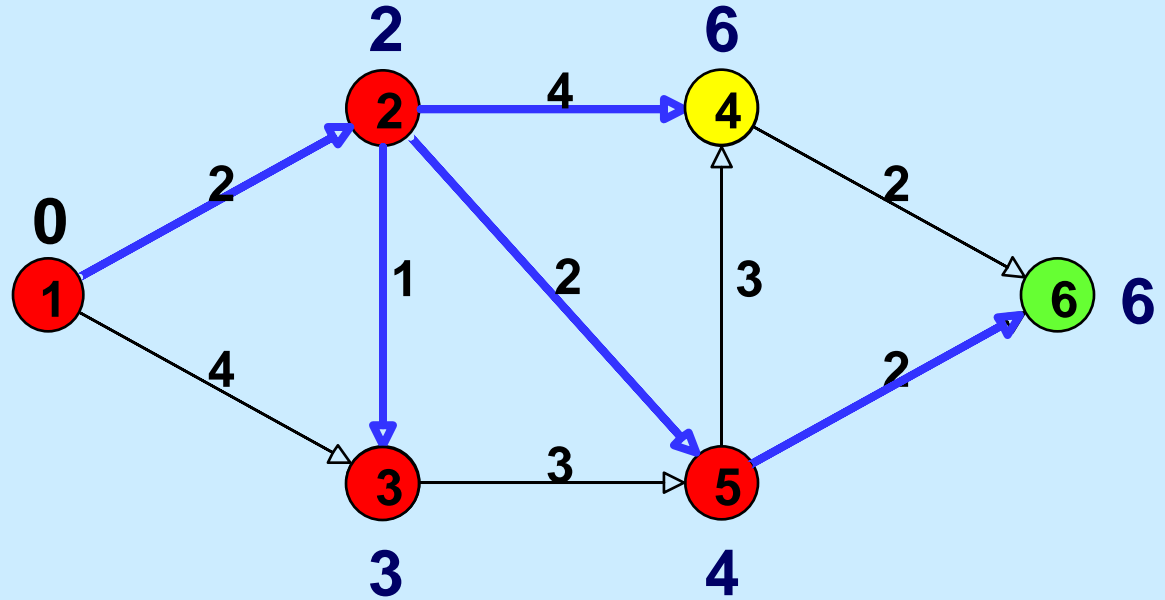




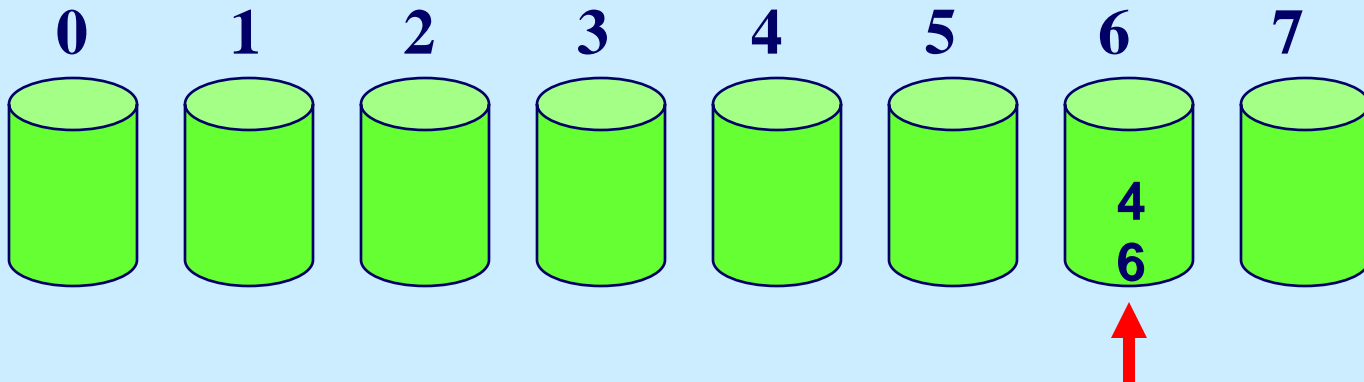
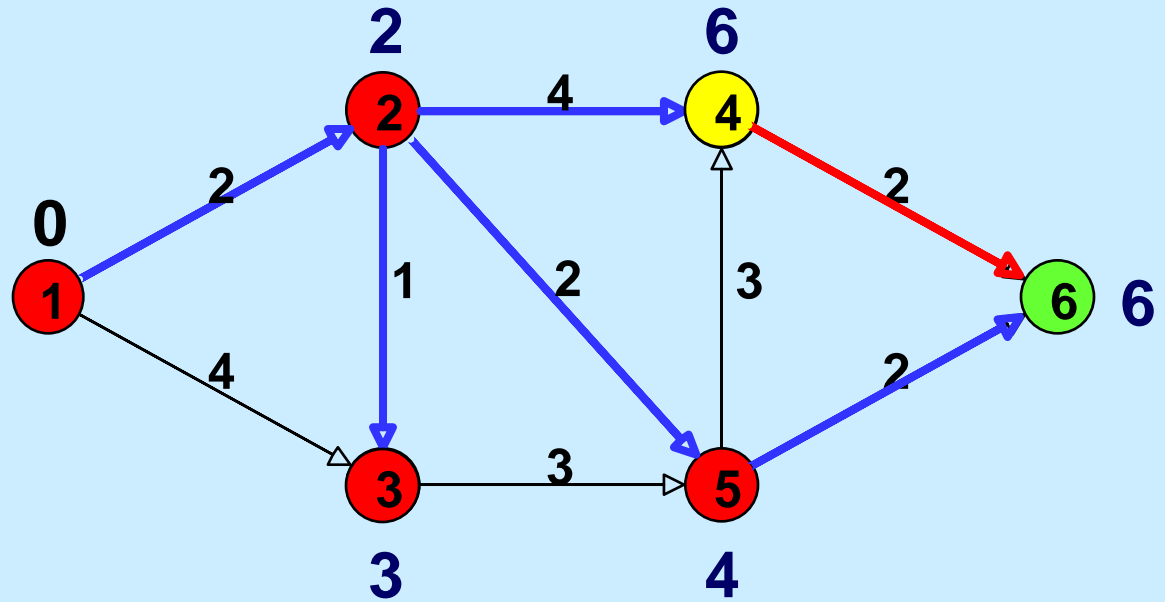
# Update



# Choose Minimum Temporary Label

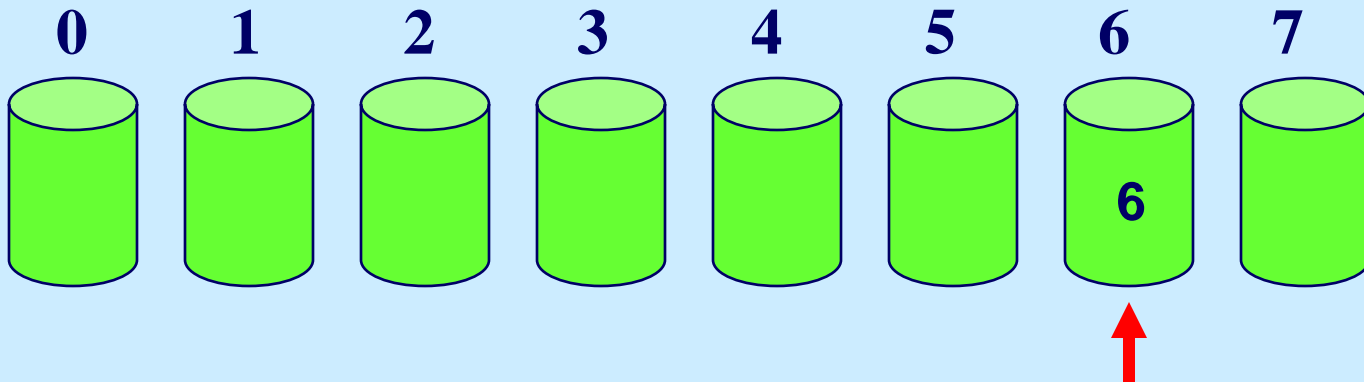
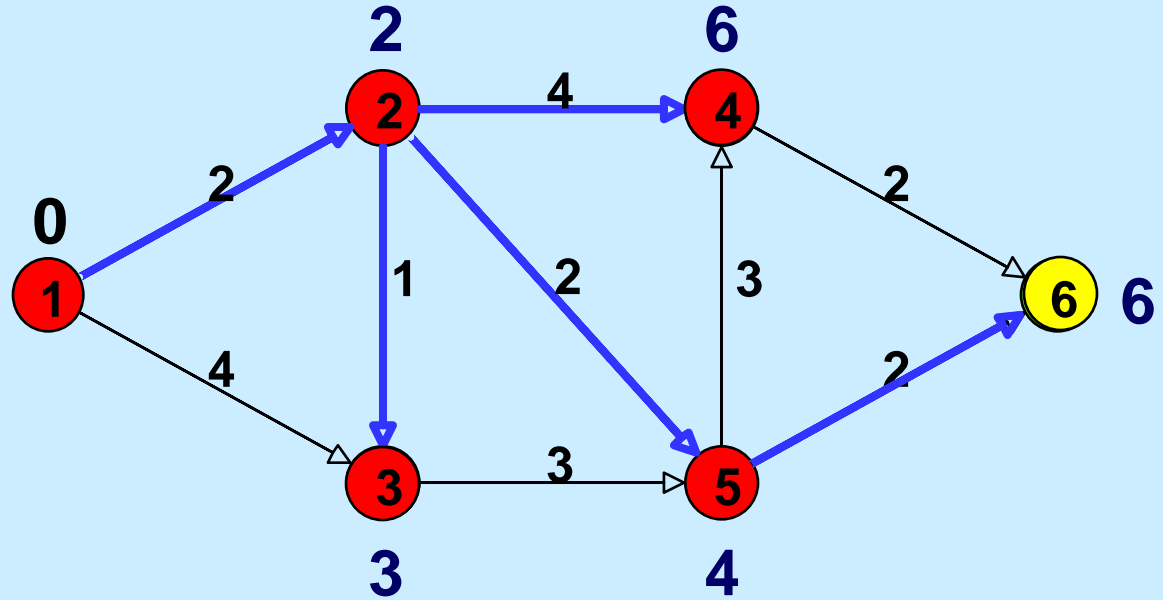


# Update



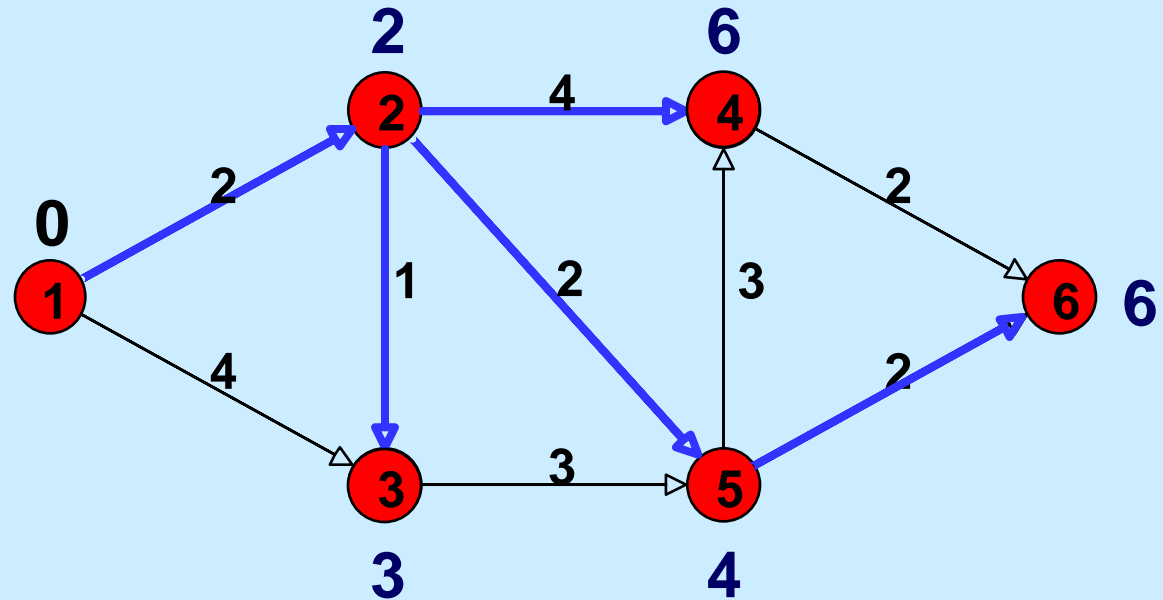
# Choose Minimum Temporary Label

There is nothing to update



# End of Algorithm

---



**All nodes are now permanent**

**The predecessors form a tree**

**The shortest path from node 1 to node 6 can be found by tracing back predecessors**

MIT OpenCourseWare  
<http://ocw.mit.edu>

15.082J / 6.855J / ESD.78J Network Optimization  
Fall 2010

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.