

The background of the slide features a scenic landscape of the Flatirons mountain range in Boulder, Colorado. The mountains are composed of light-colored, layered rock and are partially covered with green pine forests. In the foreground, there is a grassy, open field with a few small trees and shrubs. A group of people can be seen walking along a path on the left side of the field.

CSCI 1102 Computer Science 2

Meeting 26: Tuesday 5/4/2021

Dijkstra's Shortest Paths Algorithm

Review & Wrapup

EDSGER W. DIJKSTRA

Fundamental contributions to
programming as a high,
intellectual challenge.

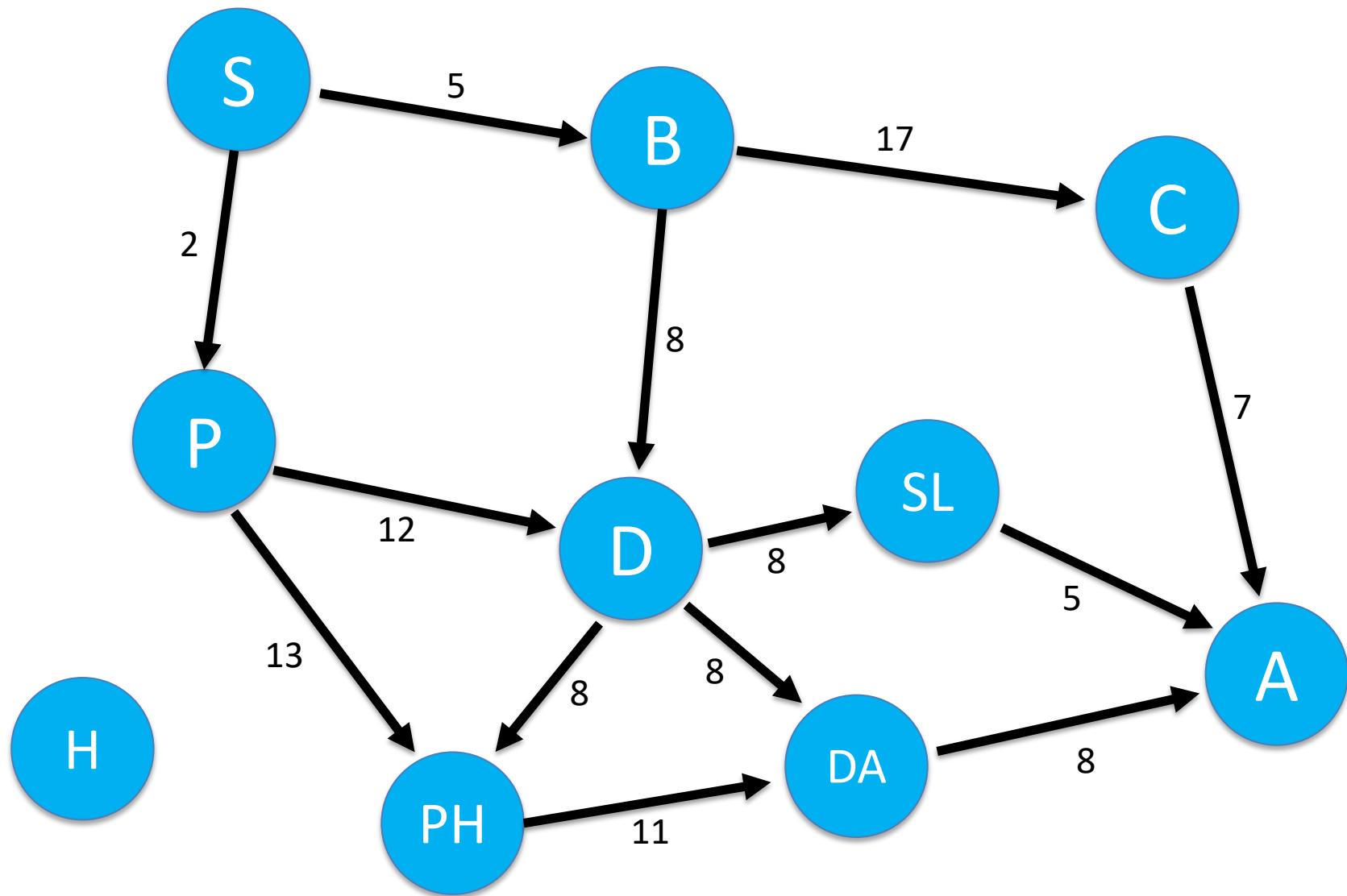


A.M.
TURING AWARD
1972

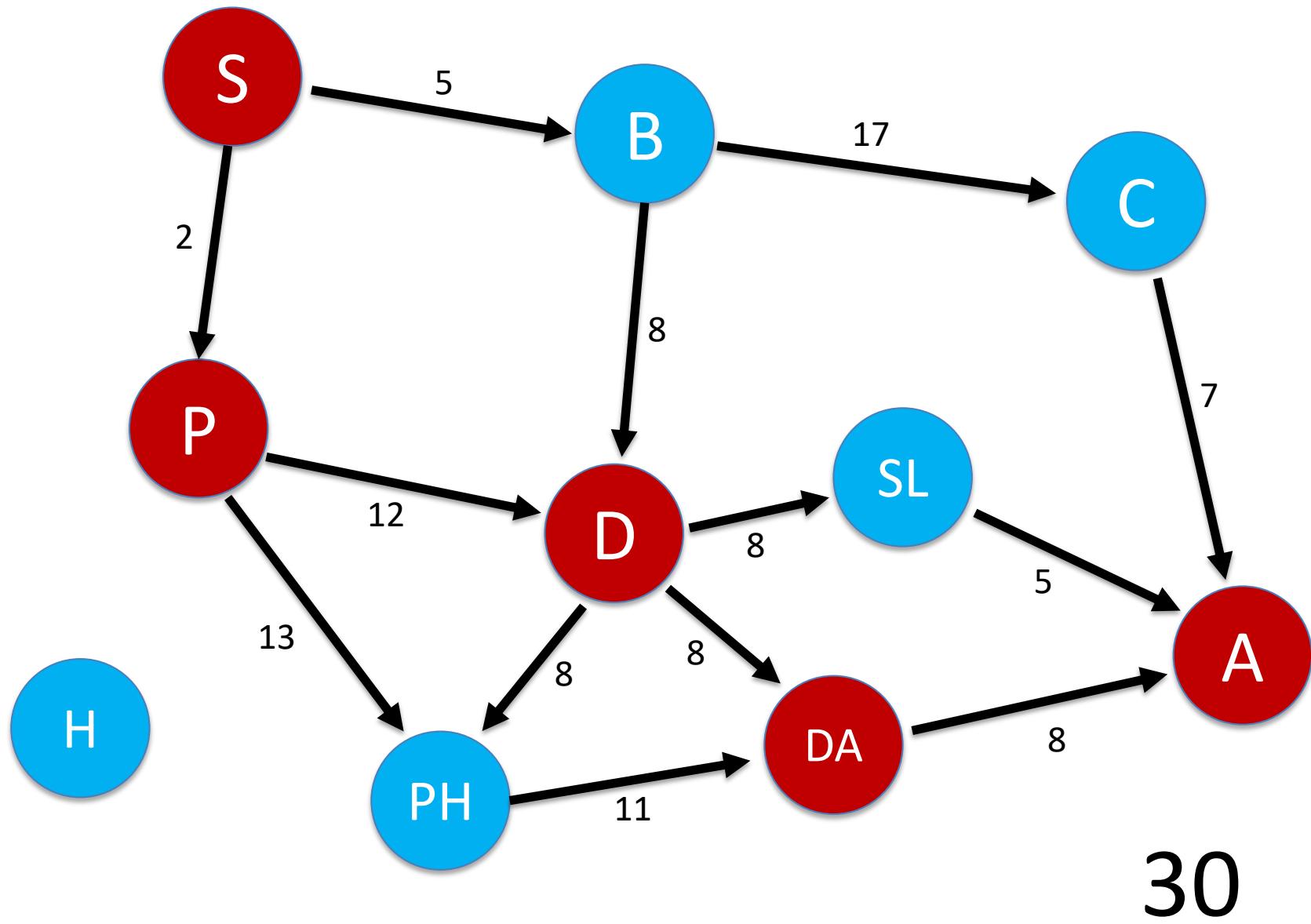


Dijkstra's Shortest Path(s) Algorithm

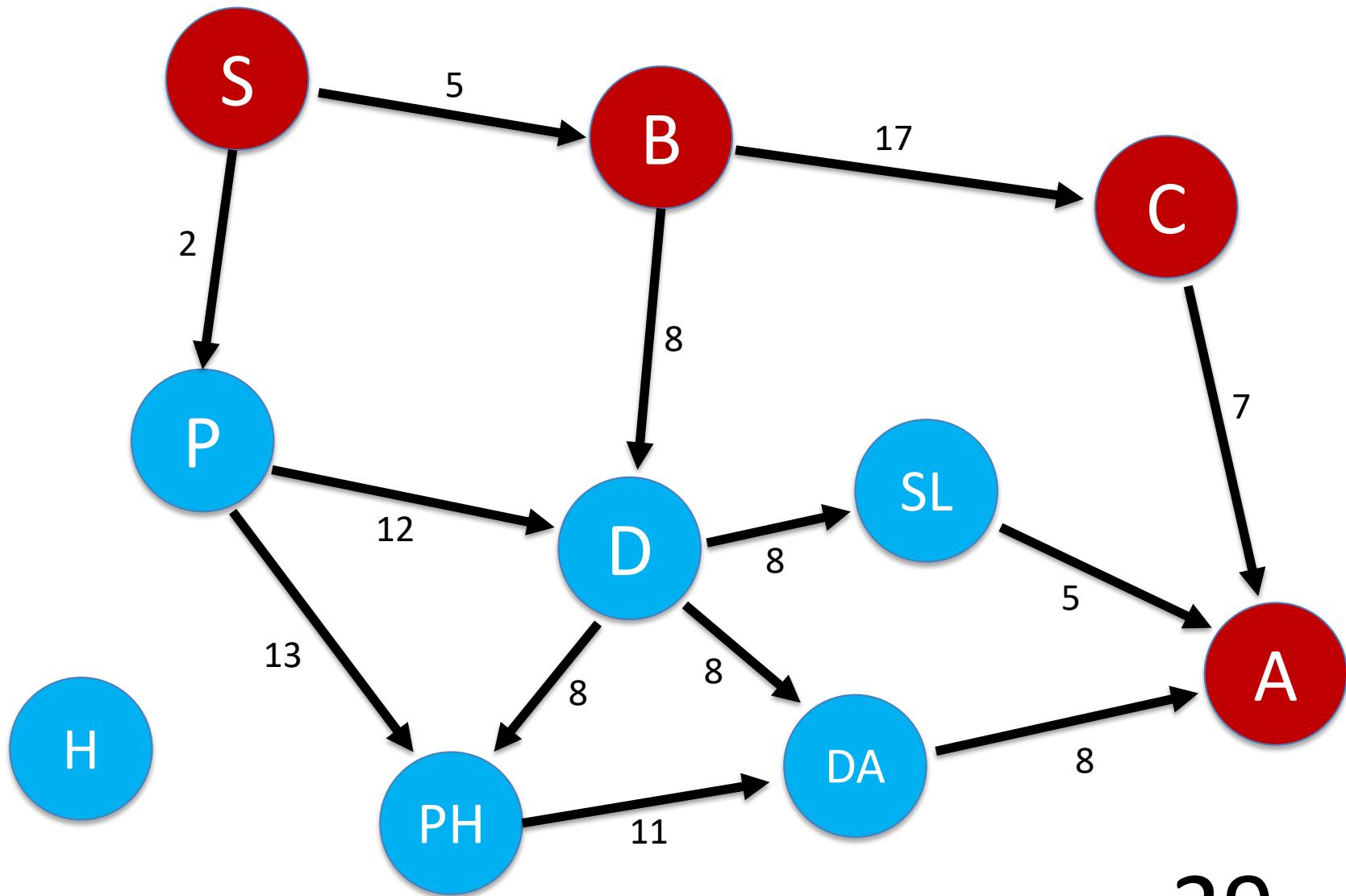
From Seattle to Atlanta



From Seattle to Atlanta

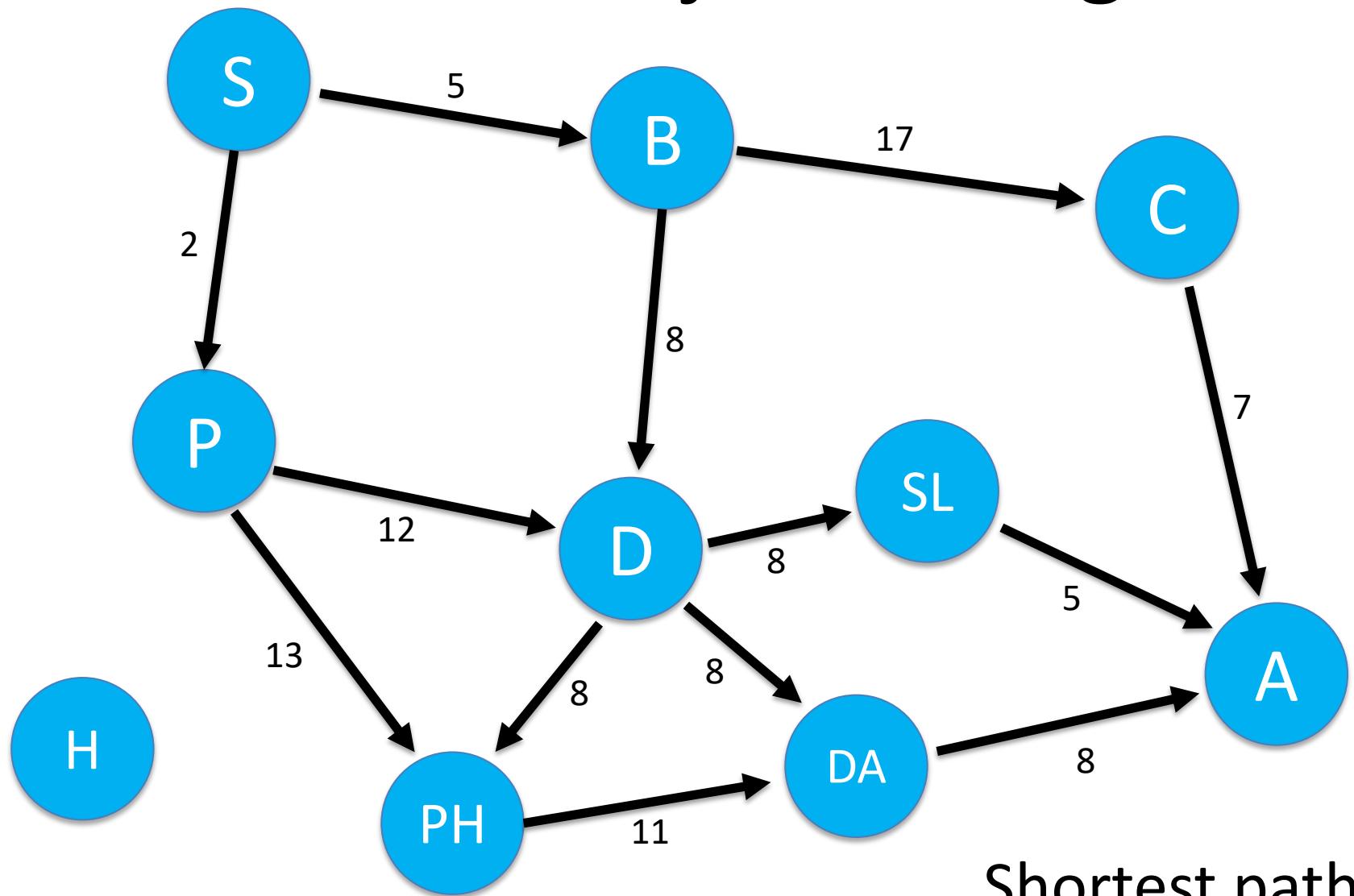


From Seattle to Atlanta



29

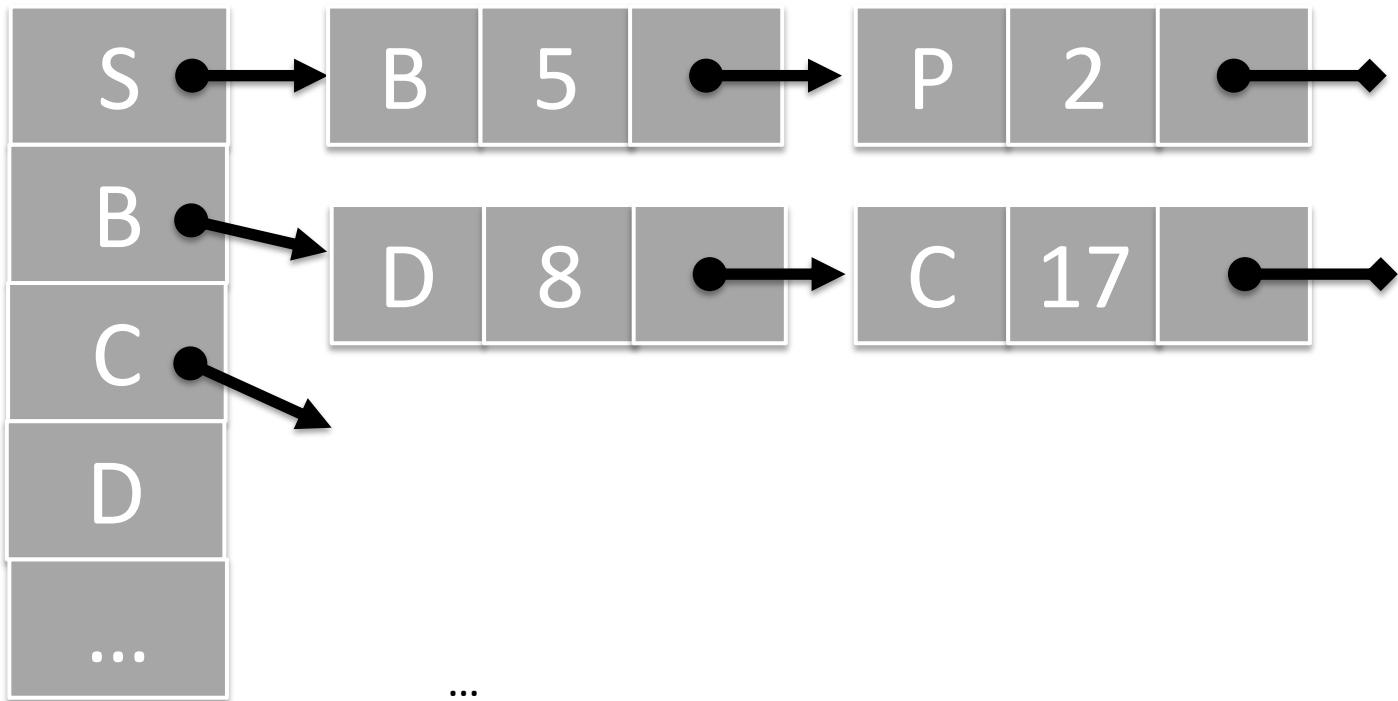
Dijkstra's Algorithm



Shortest paths
to all vertices

Adjacency

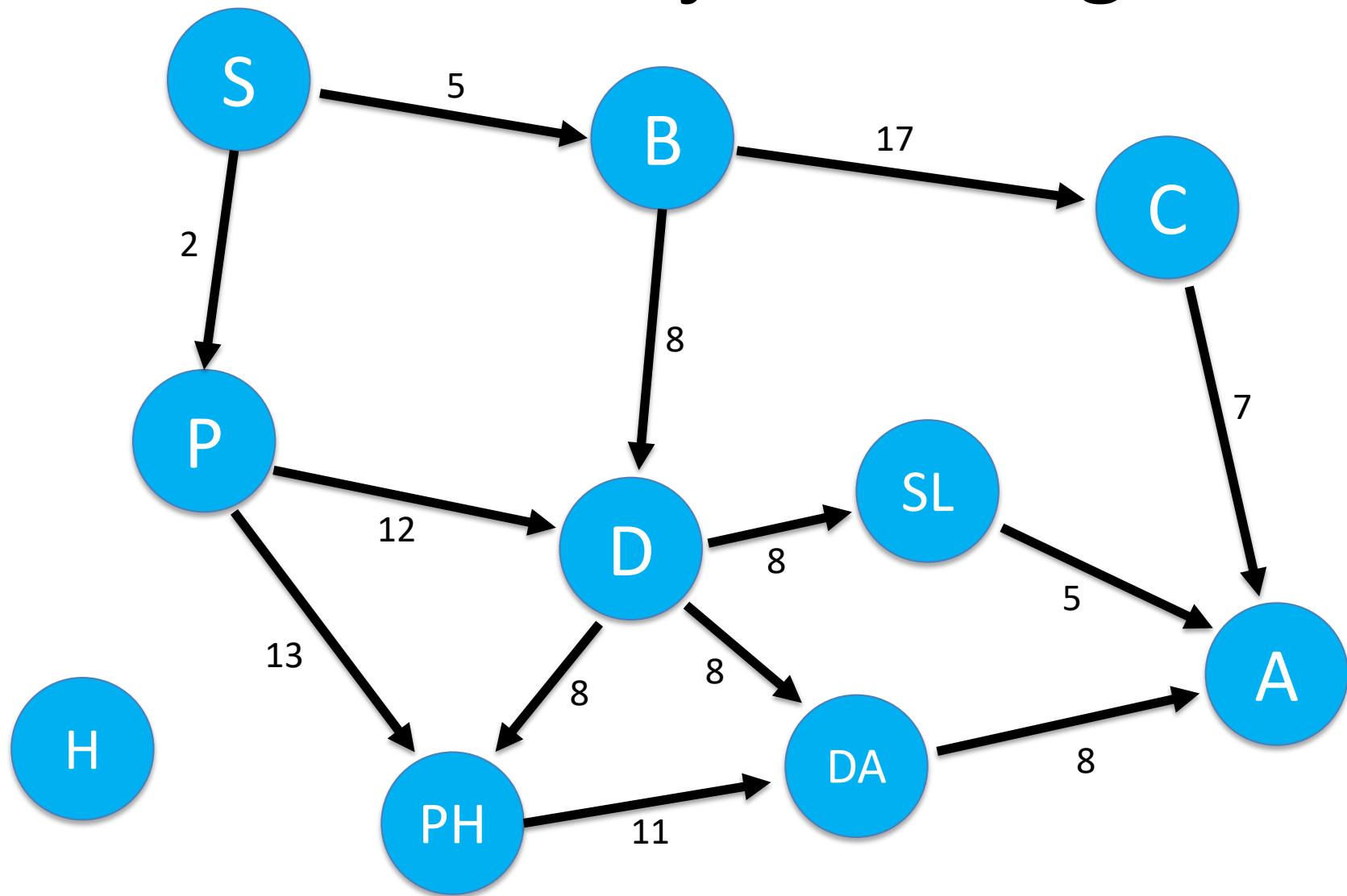
List



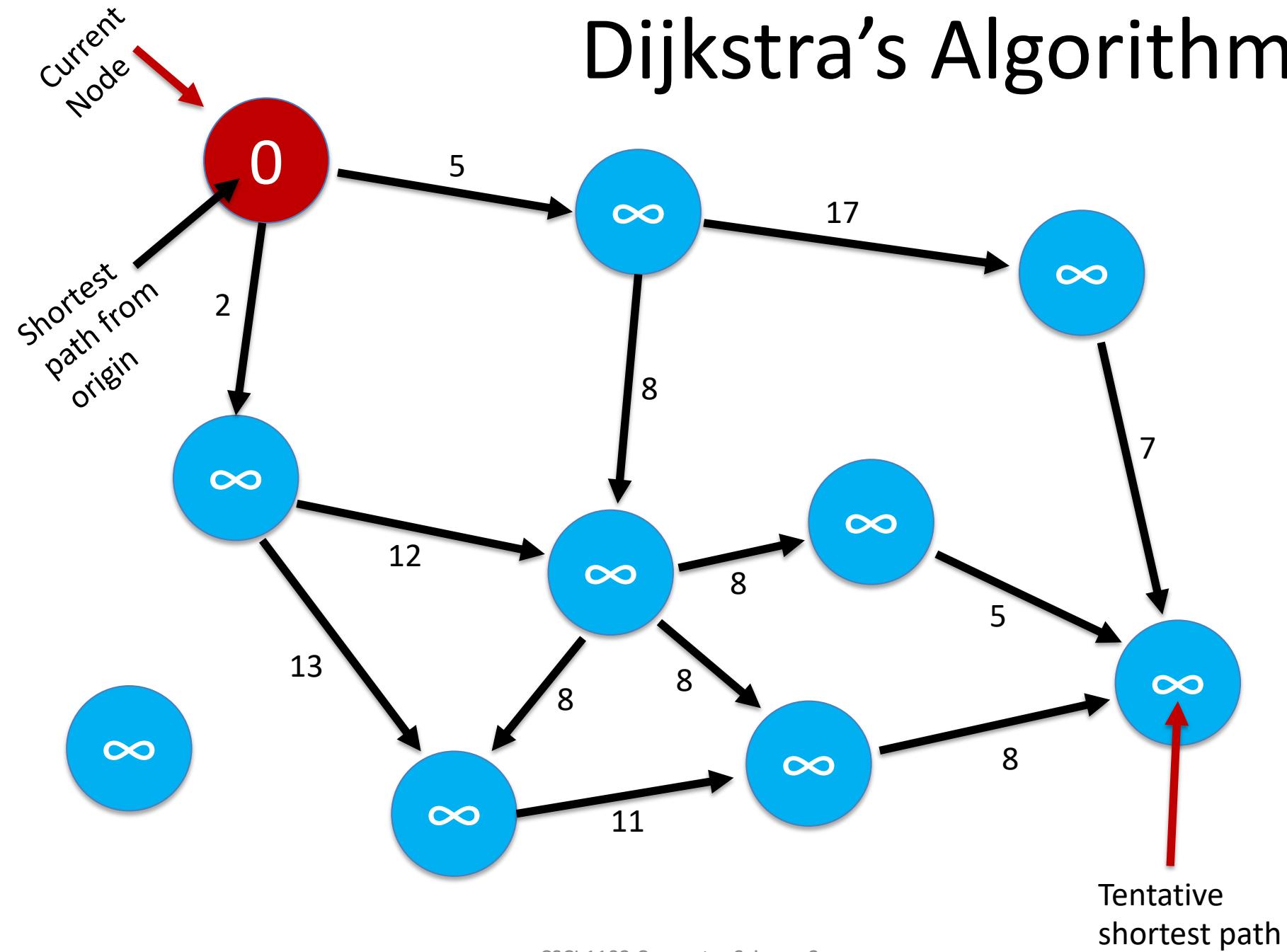
Adjacency Matrix

	S	B	P	D	C
S	0	5	2	-	-
B	-	0	-	8	17
P	-	-	0	12	-
D	-	-	-	0	-
C	-	-	-	-	0

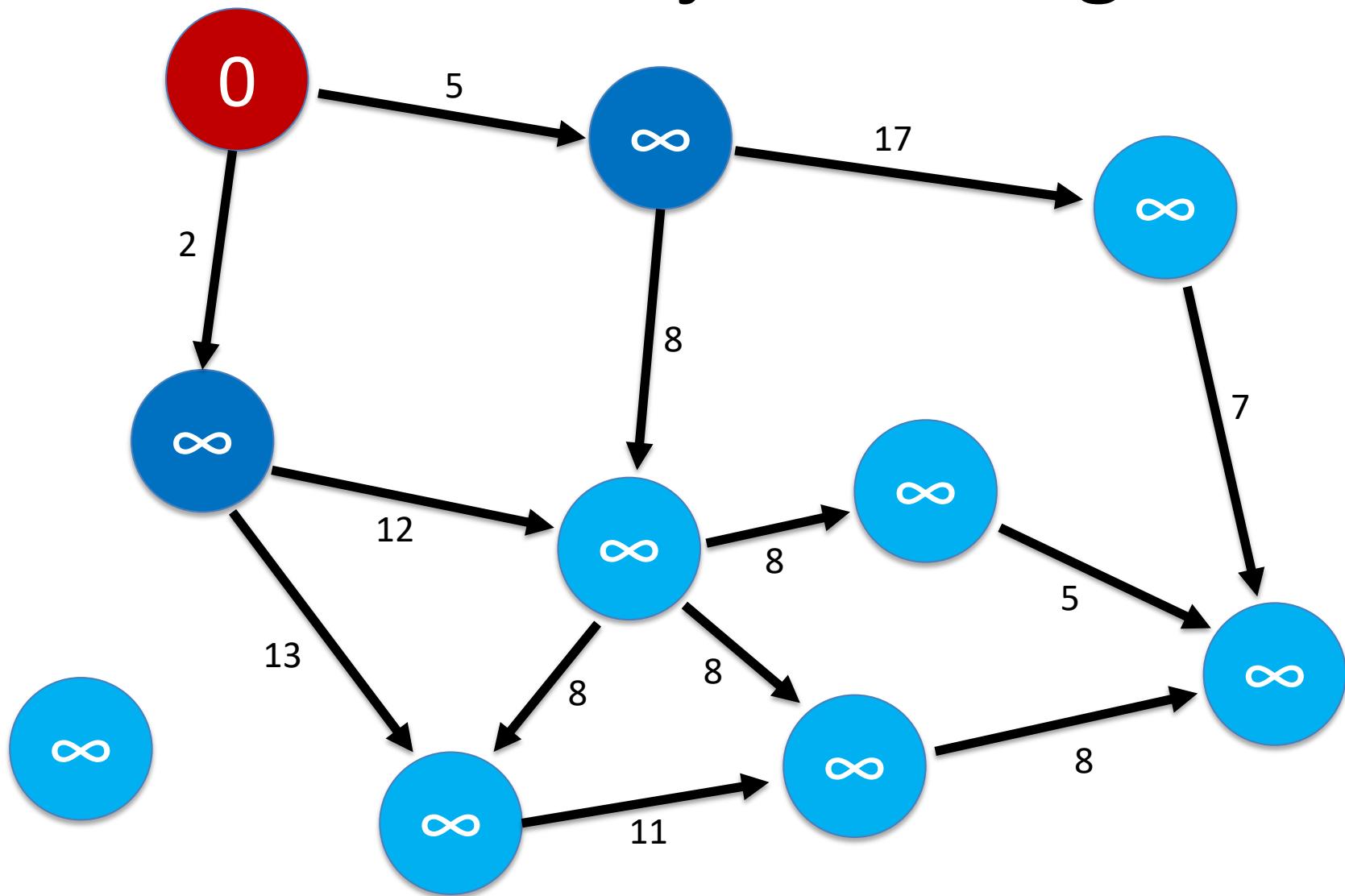
Dijkstra's Algorithm



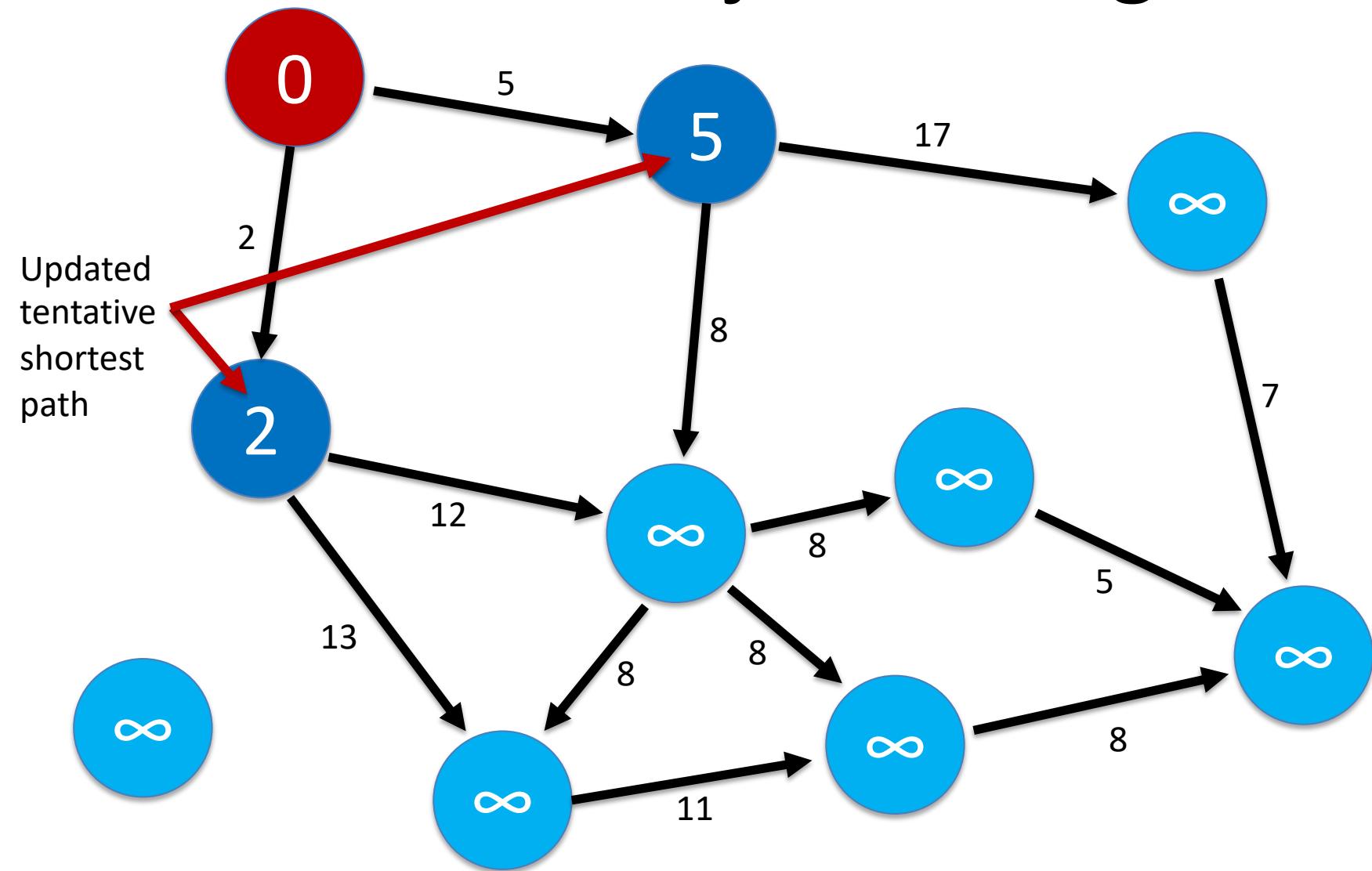
Dijkstra's Algorithm



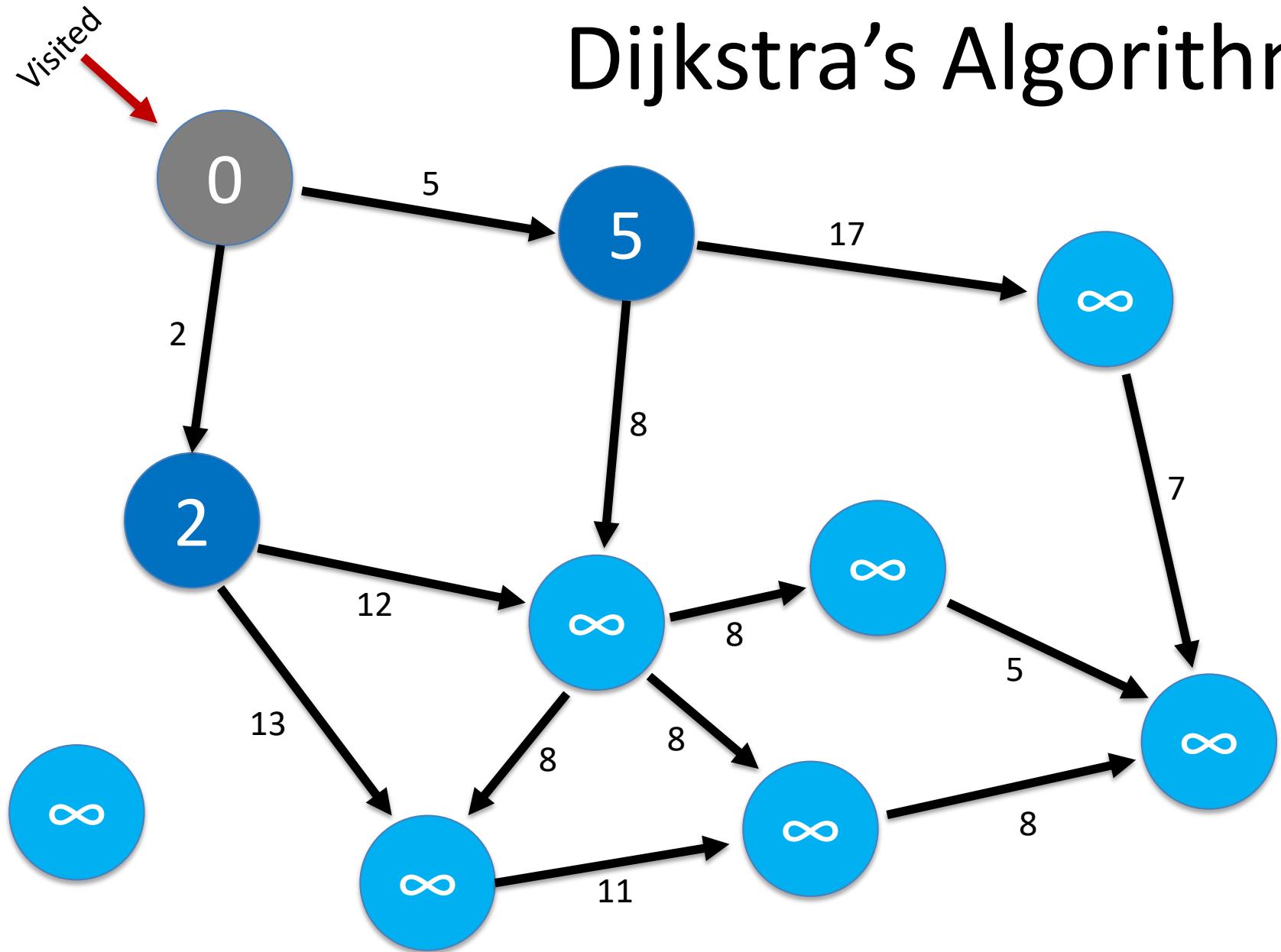
Dijkstra's Algorithm



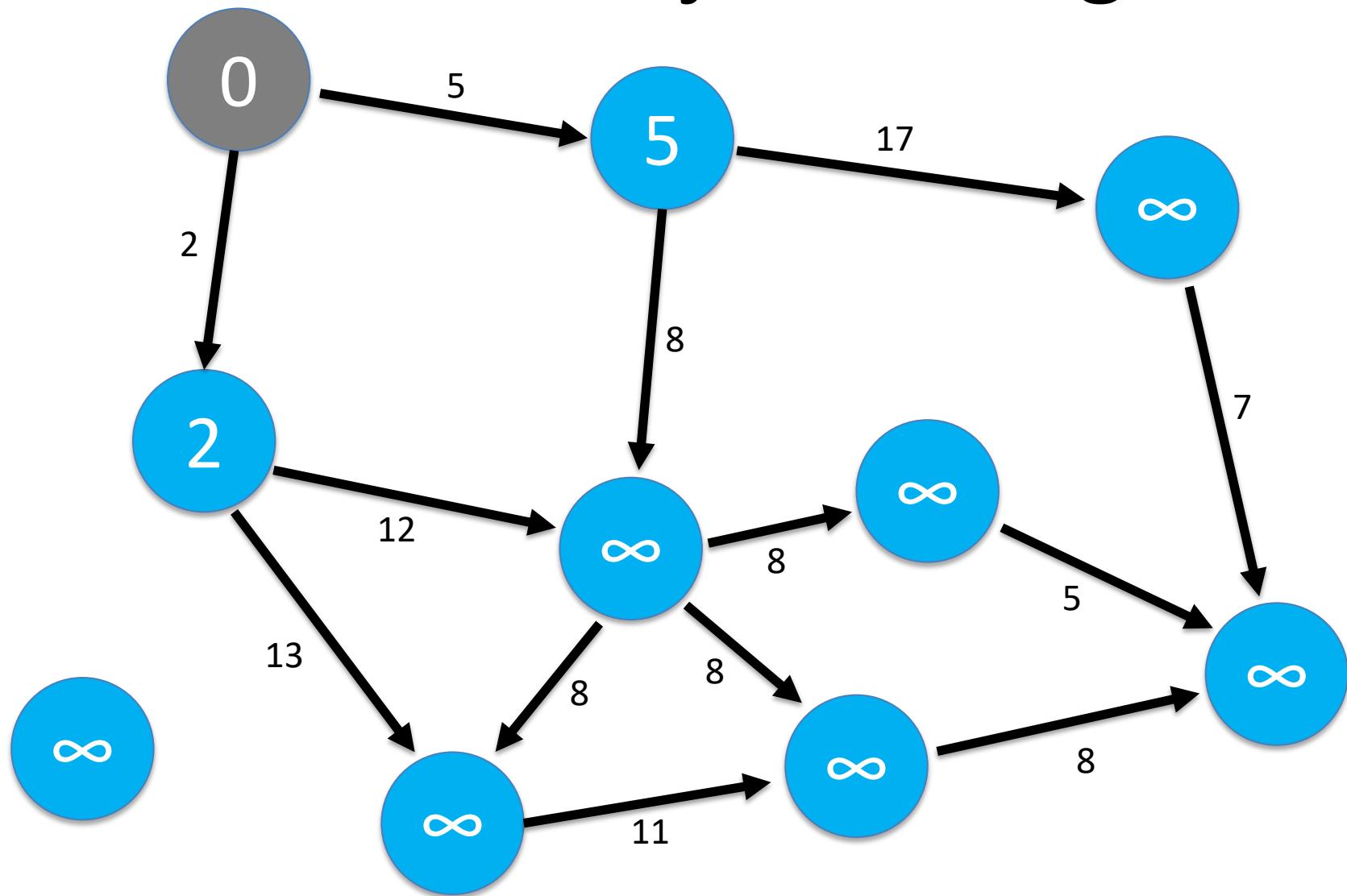
Dijkstra's Algorithm



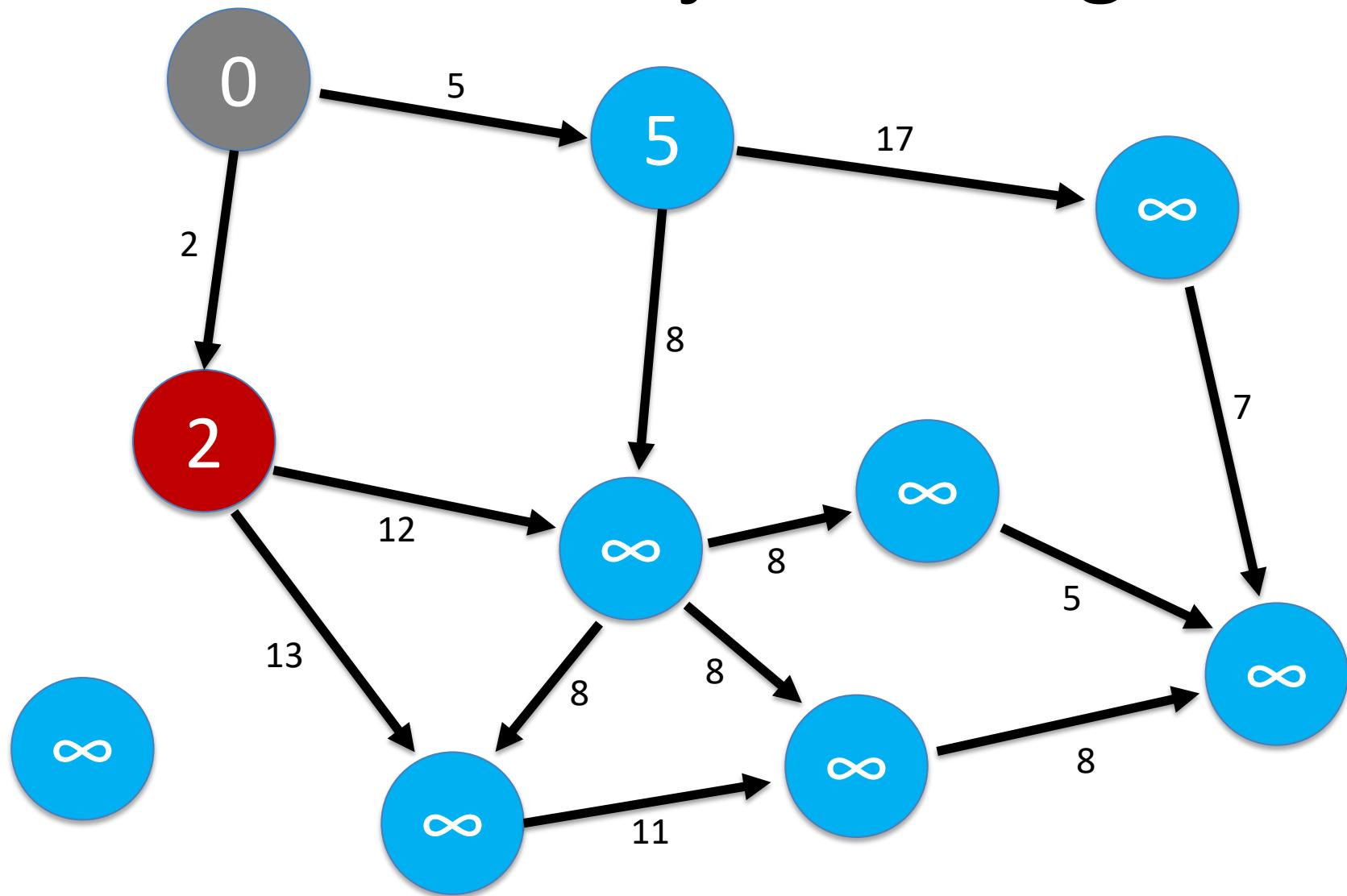
Dijkstra's Algorithm



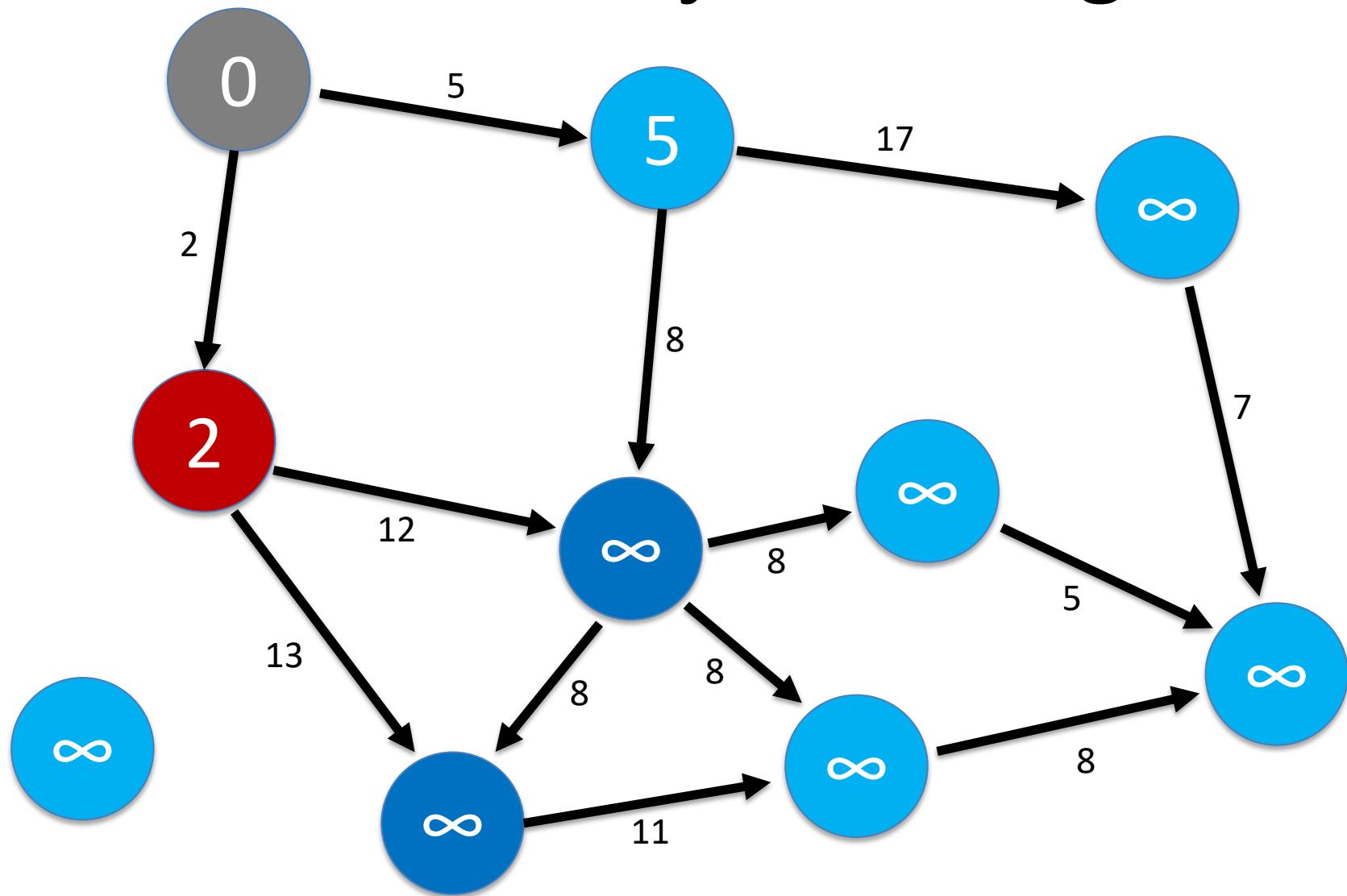
Dijkstra's Algorithm



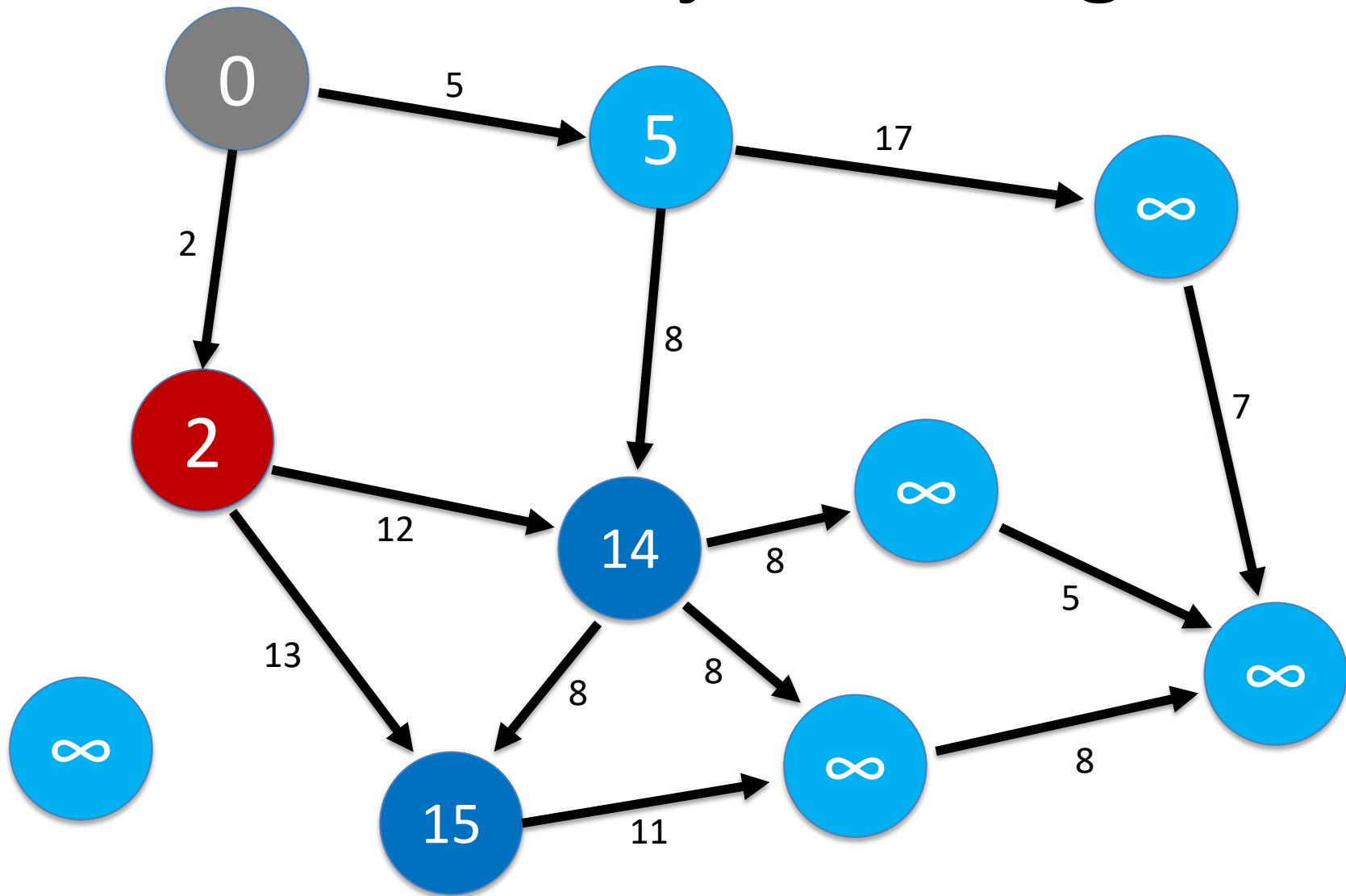
Dijkstra's Algorithm



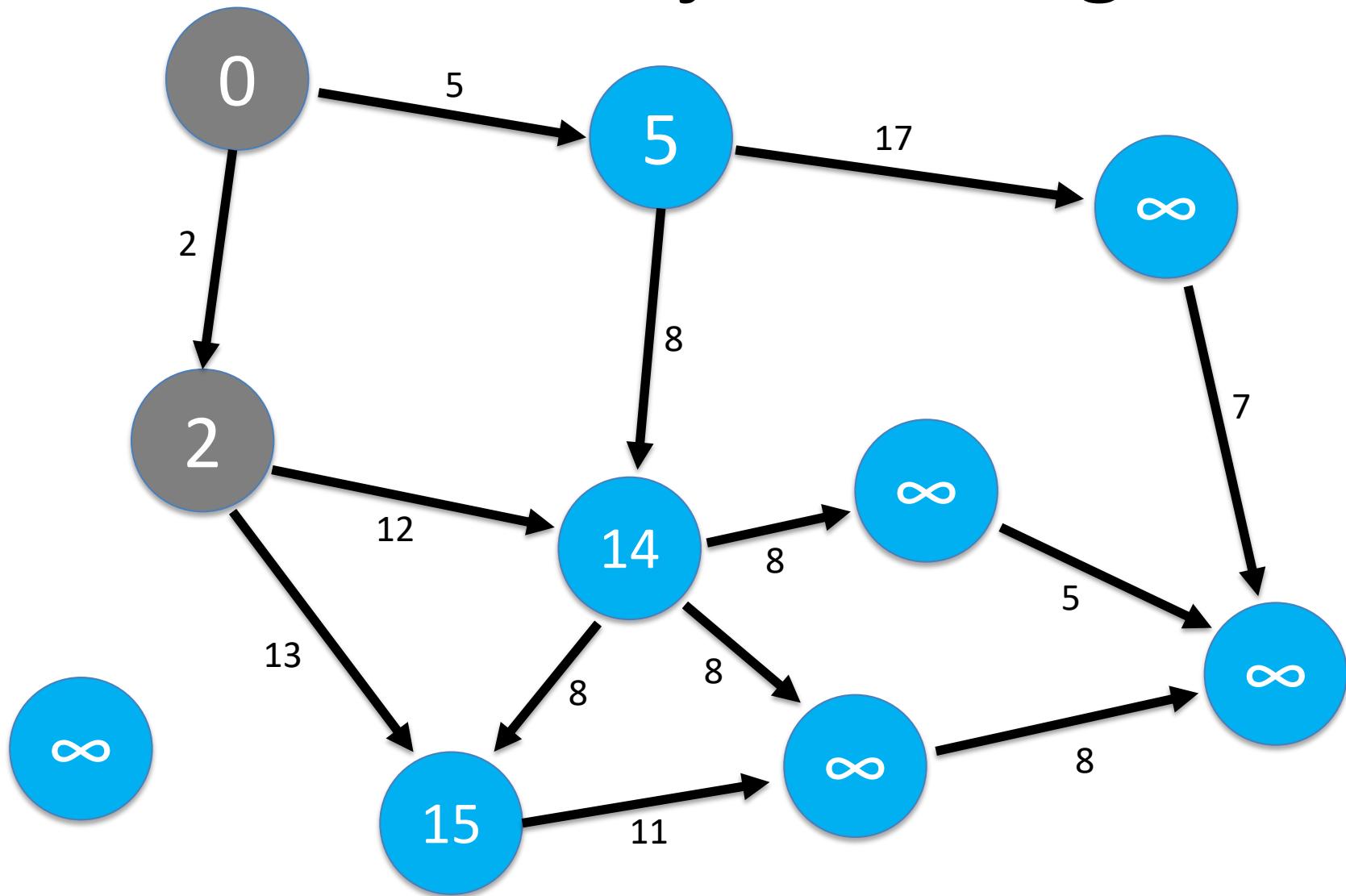
Dijkstra's Algorithm



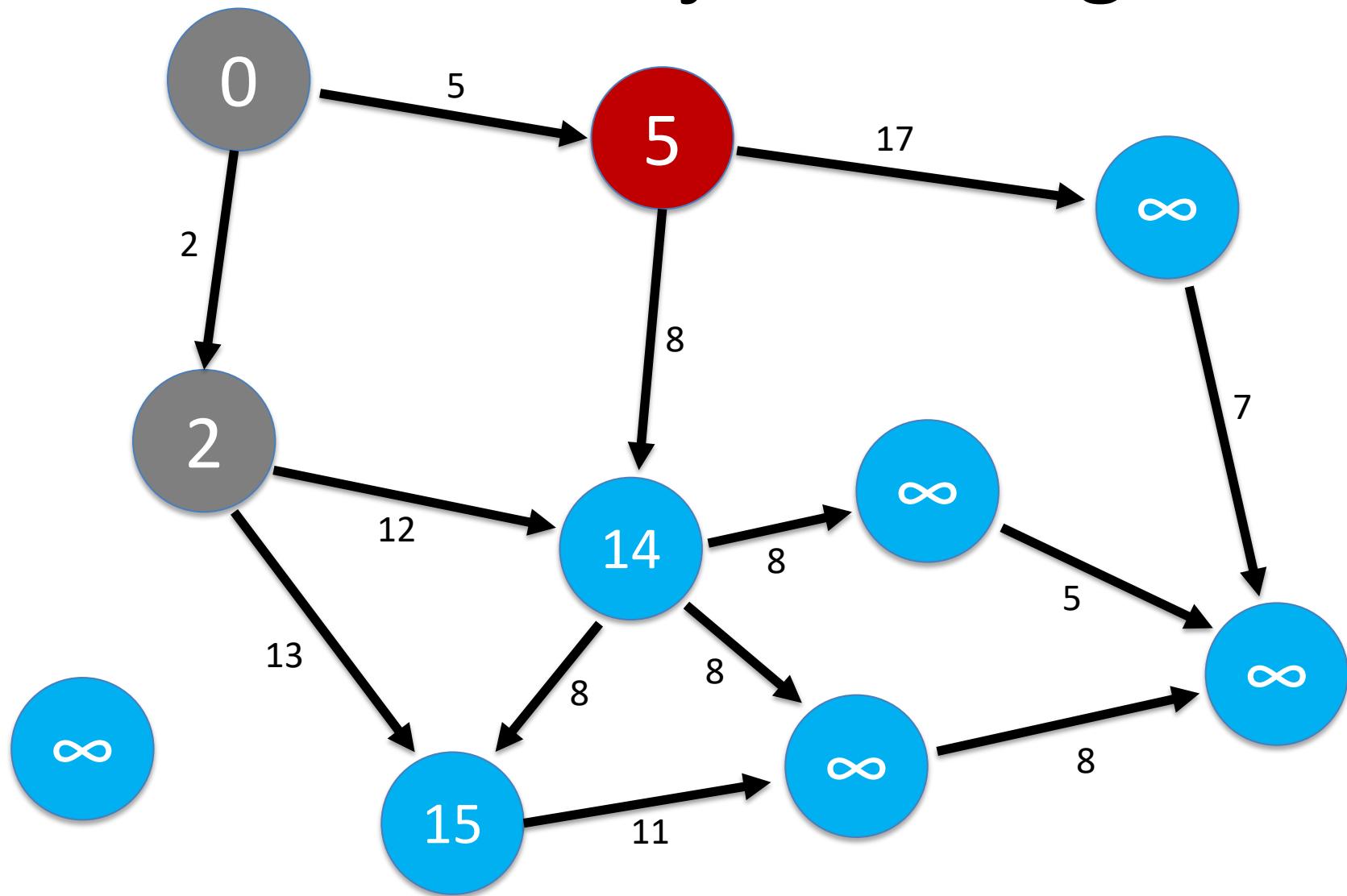
Dijkstra's Algorithm



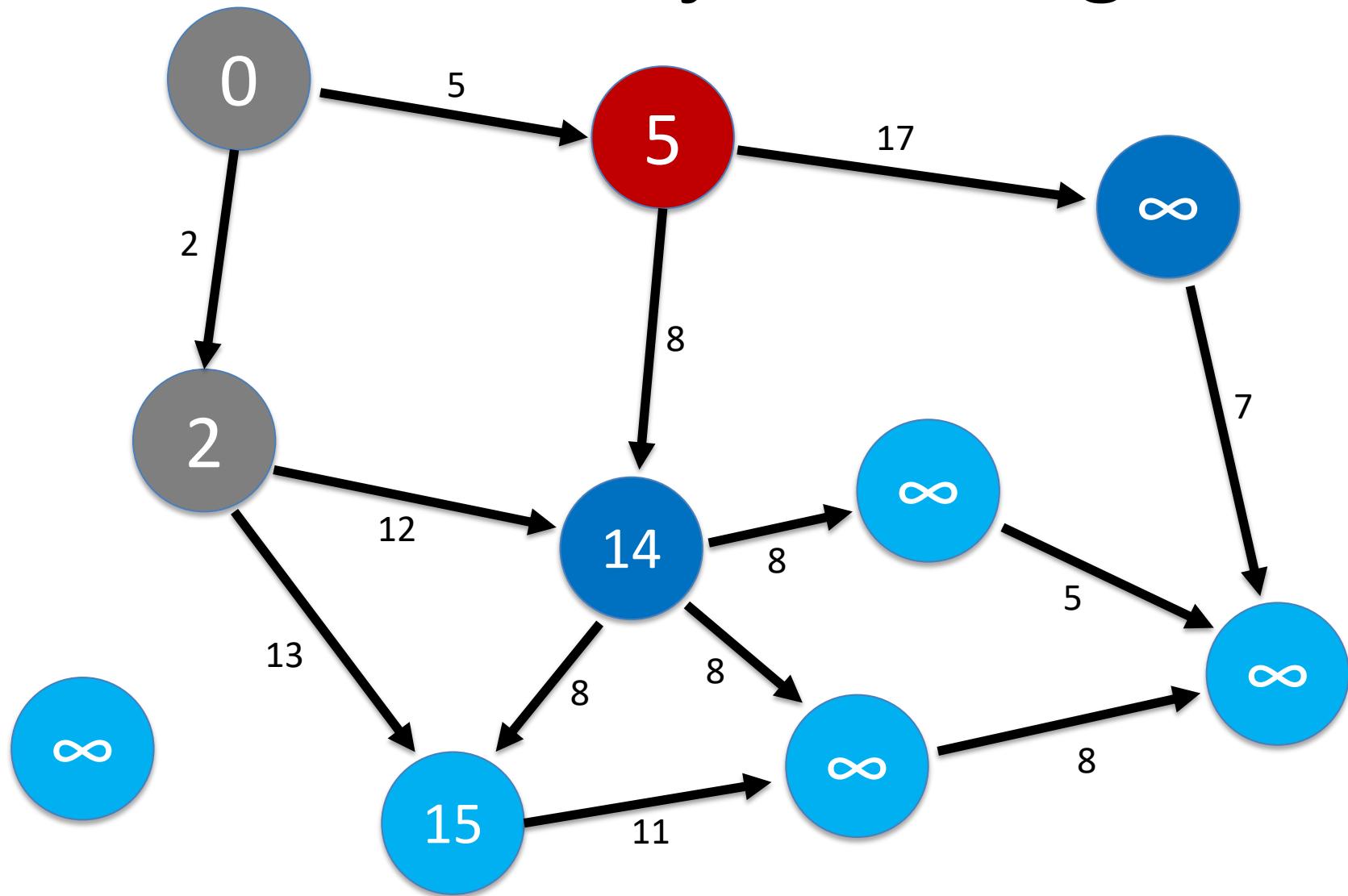
Dijkstra's Algorithm



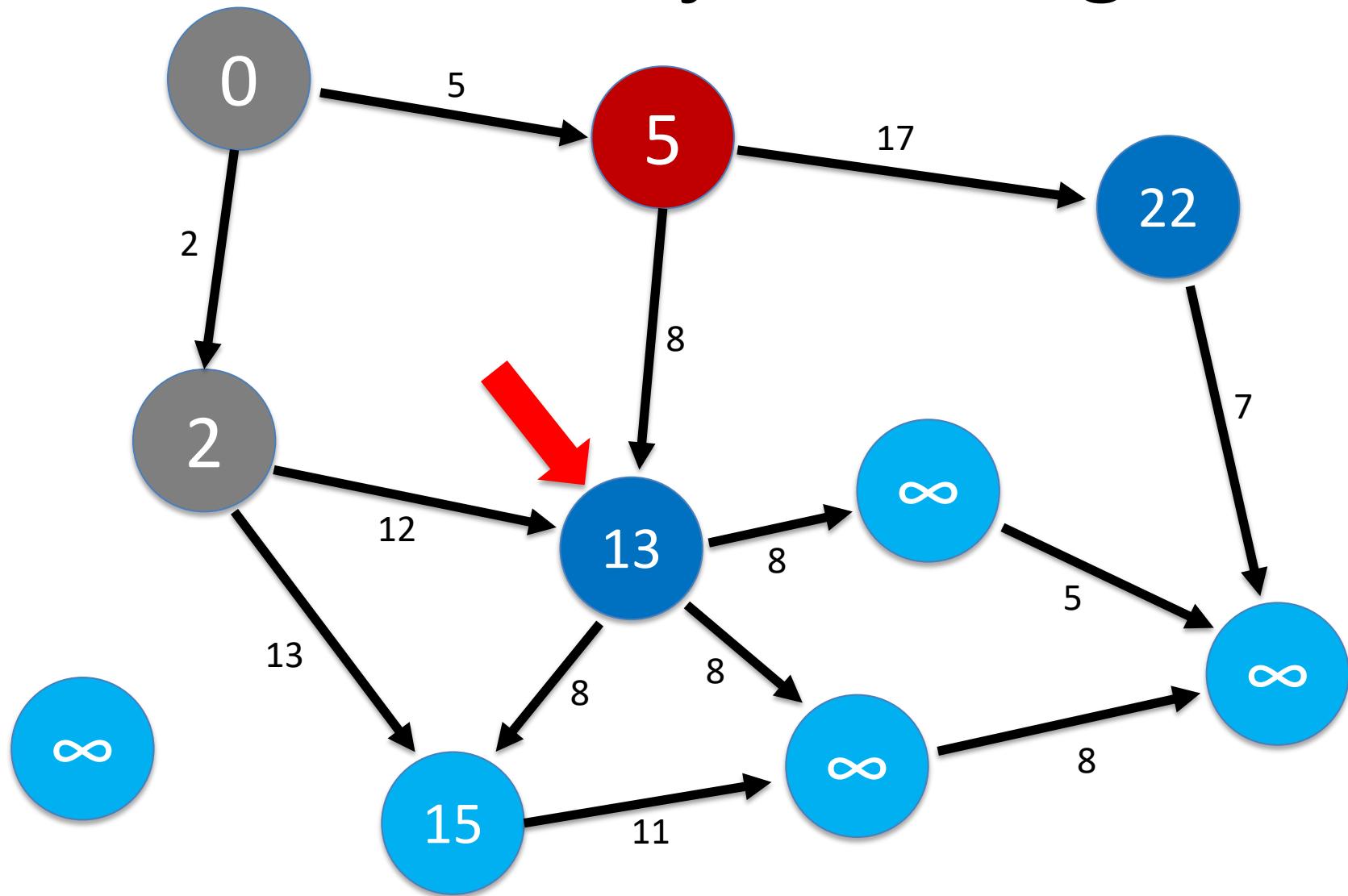
Dijkstra's Algorithm



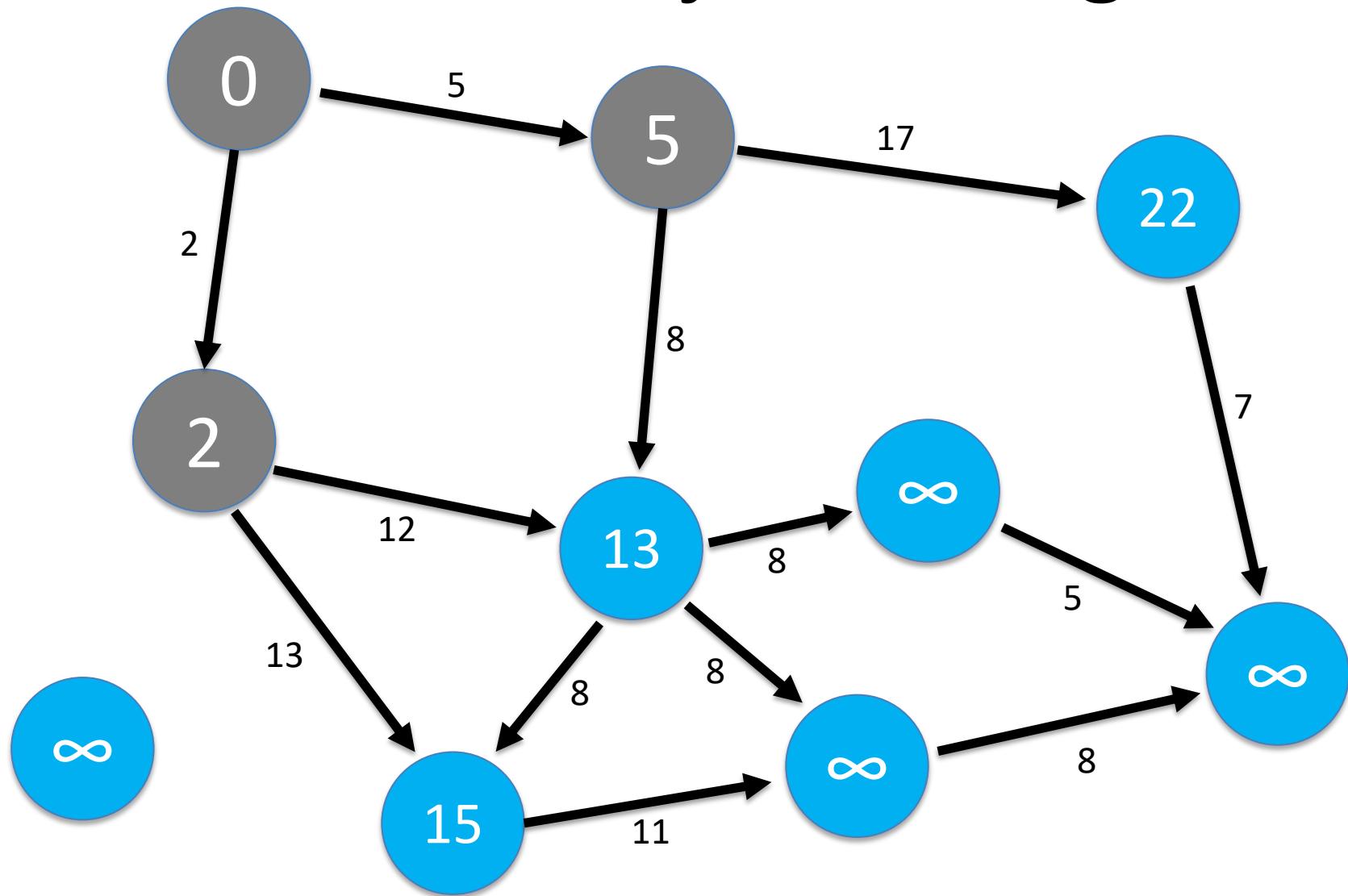
Dijkstra's Algorithm



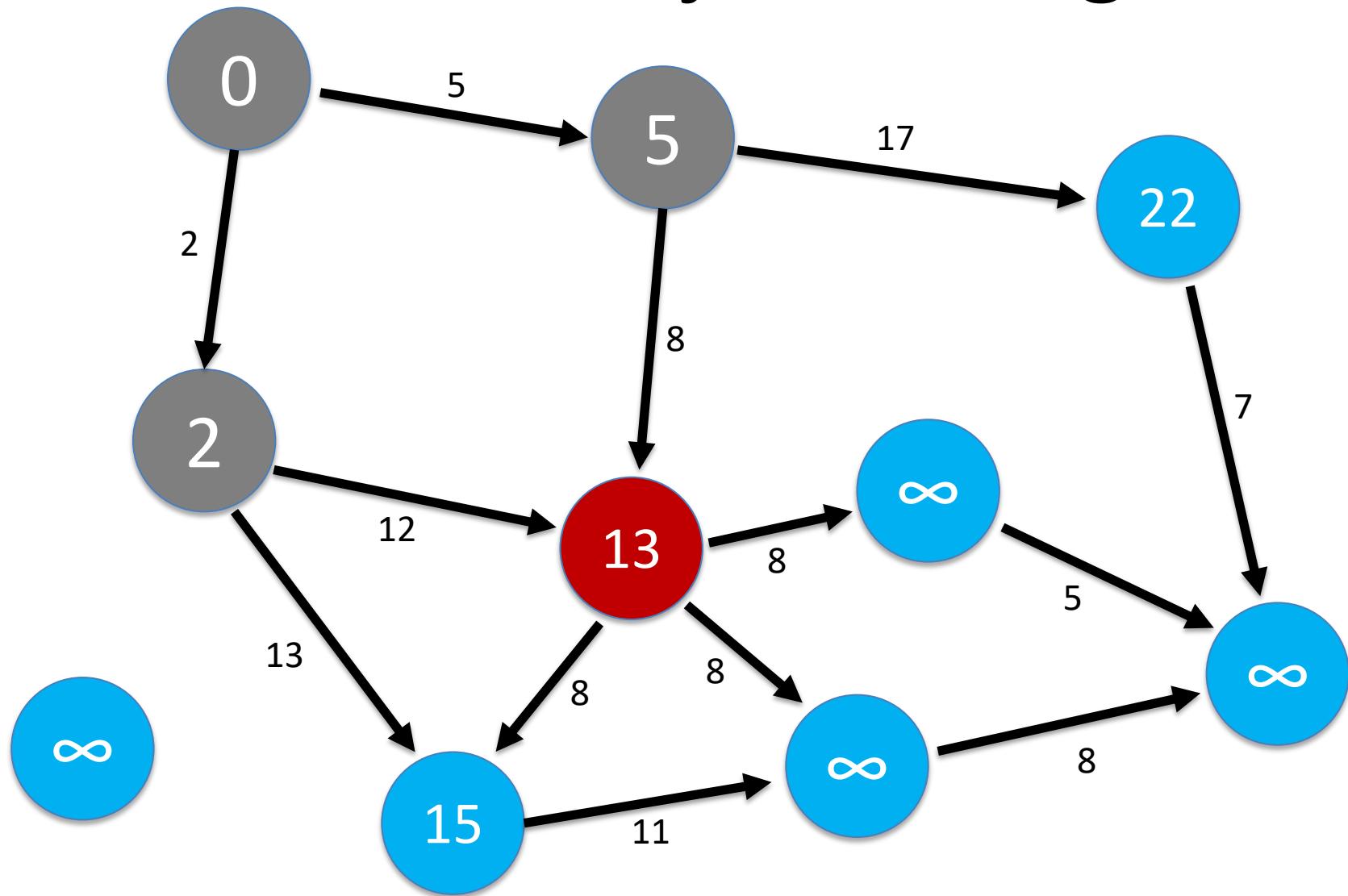
Dijkstra's Algorithm



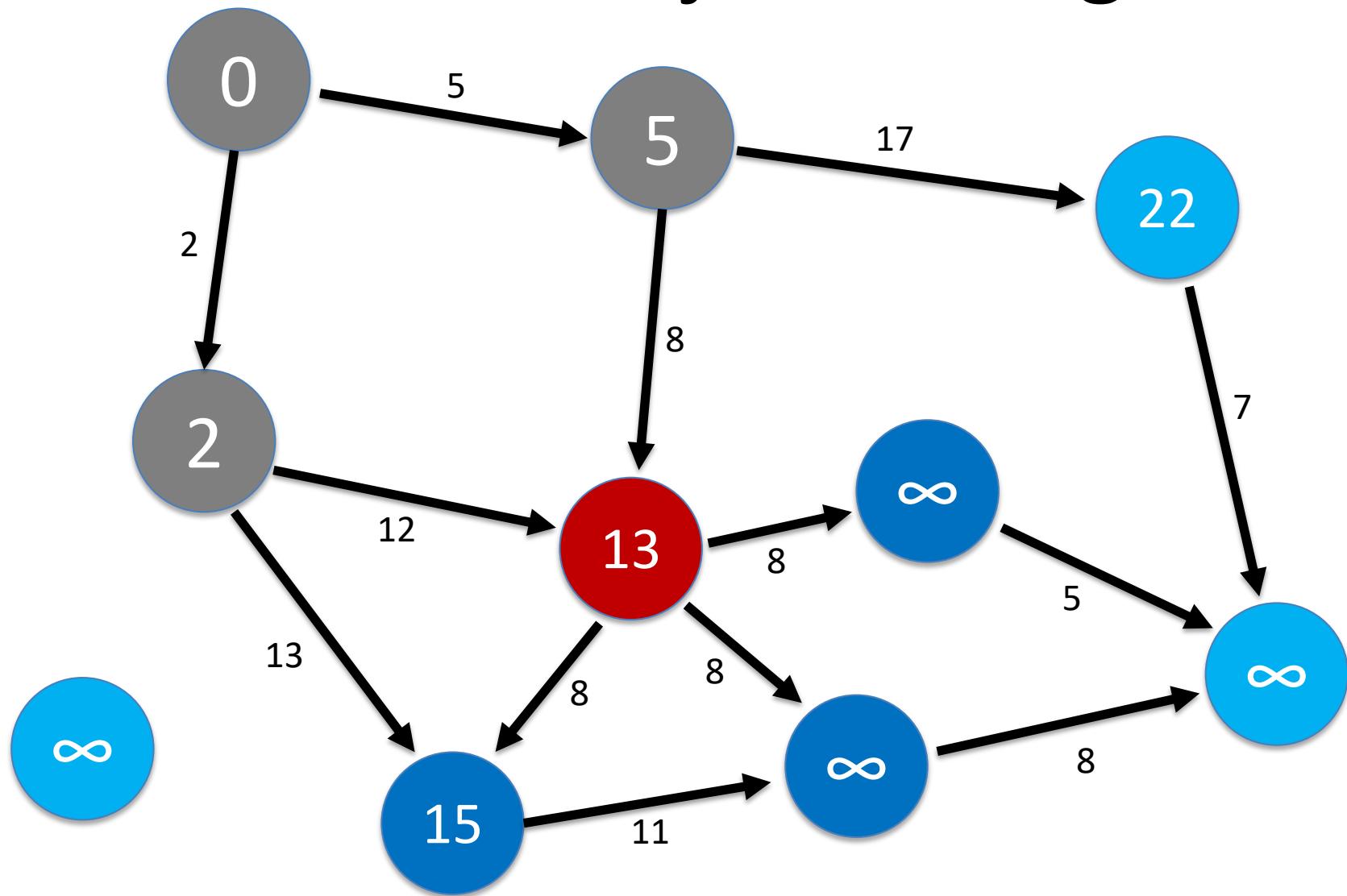
Dijkstra's Algorithm



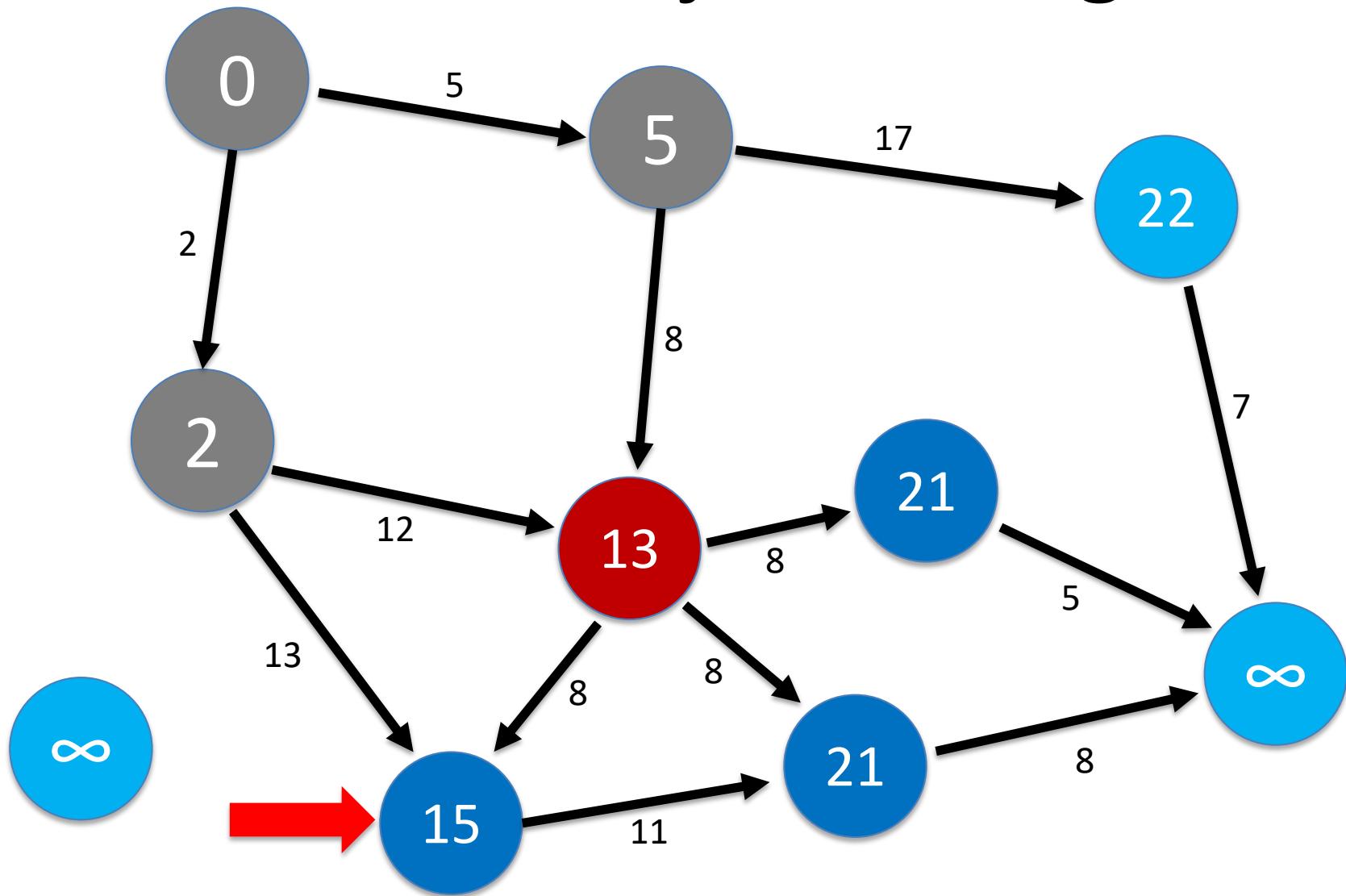
Dijkstra's Algorithm



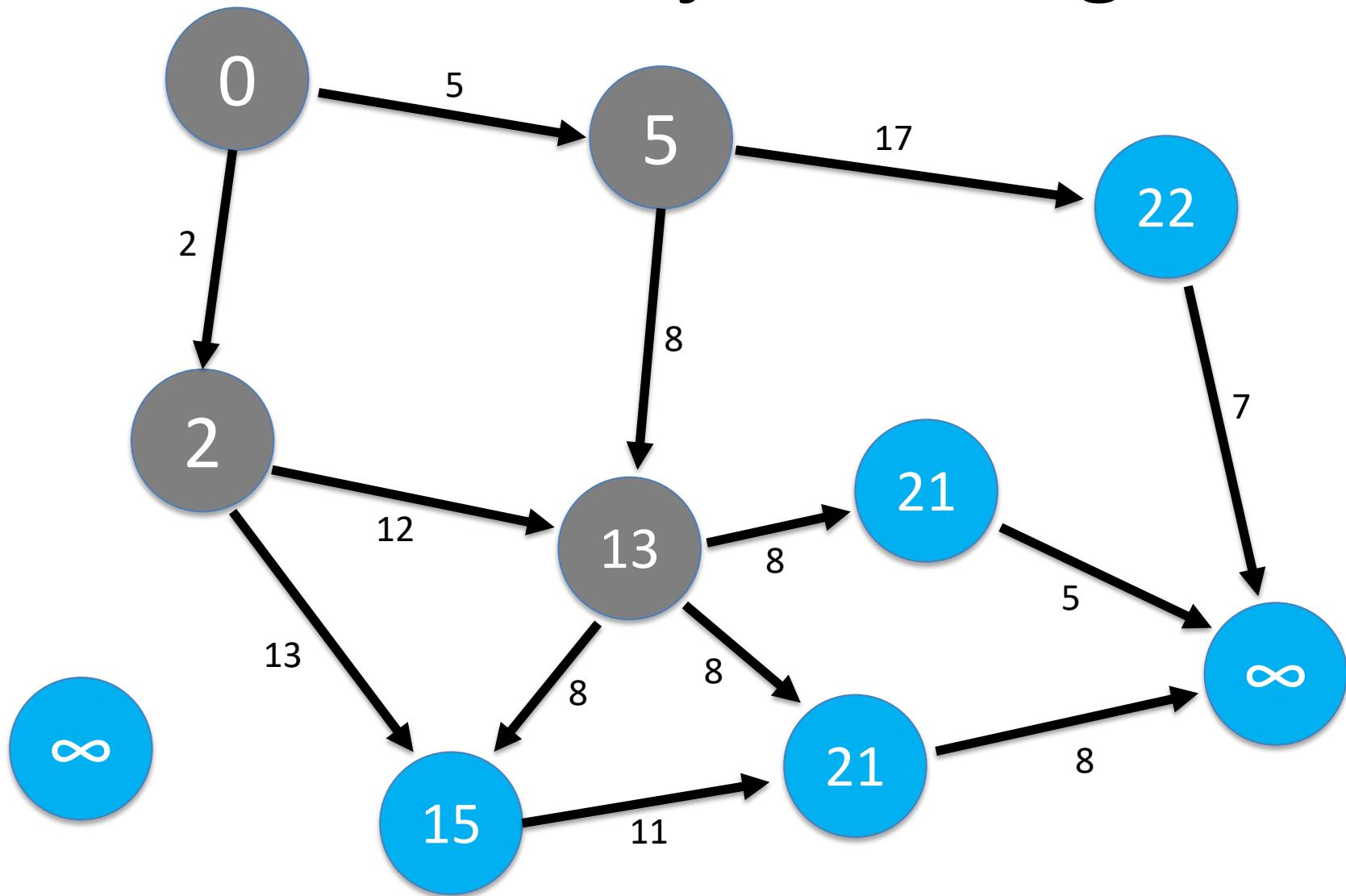
Dijkstra's Algorithm



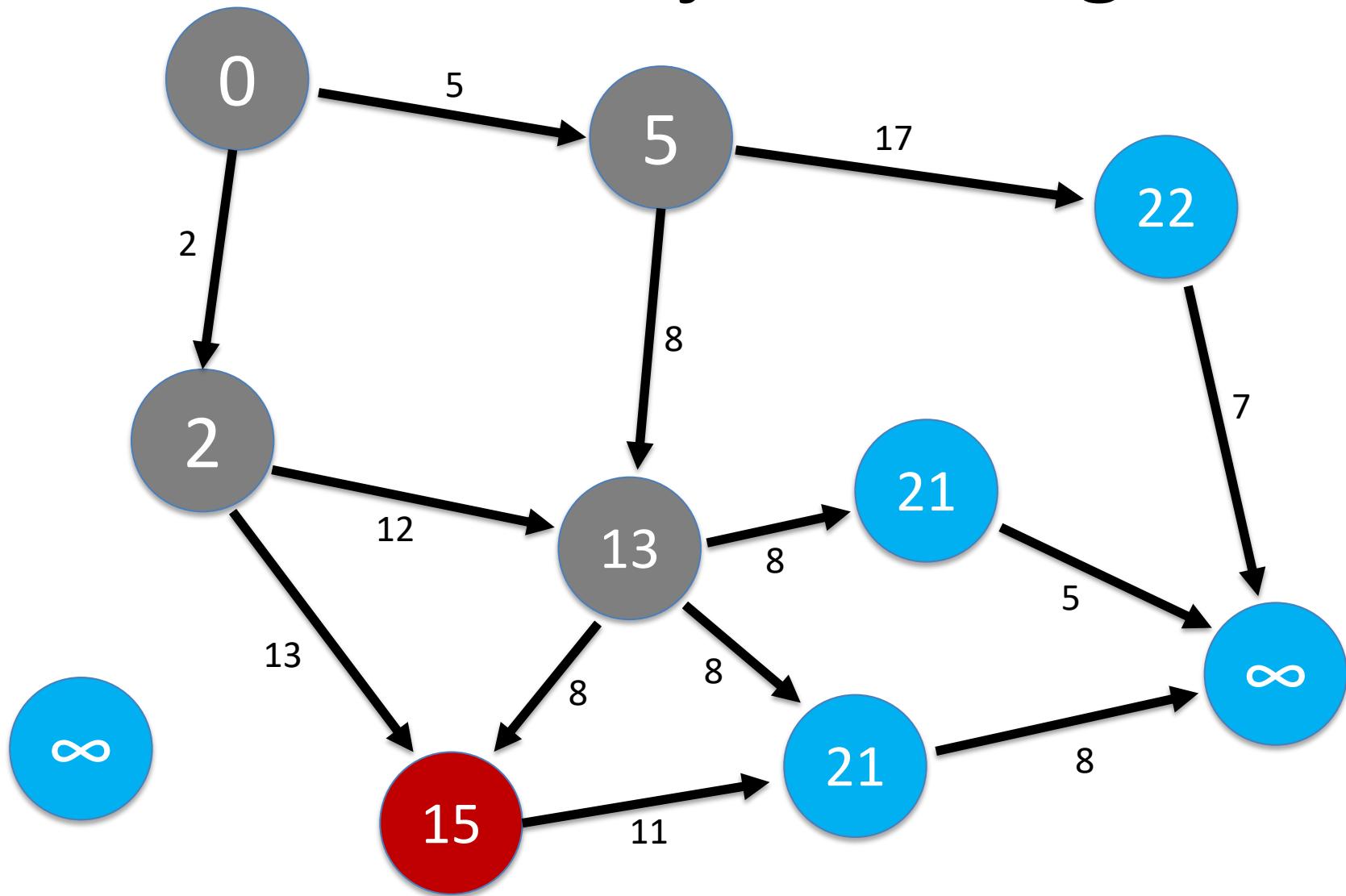
Dijkstra's Algorithm



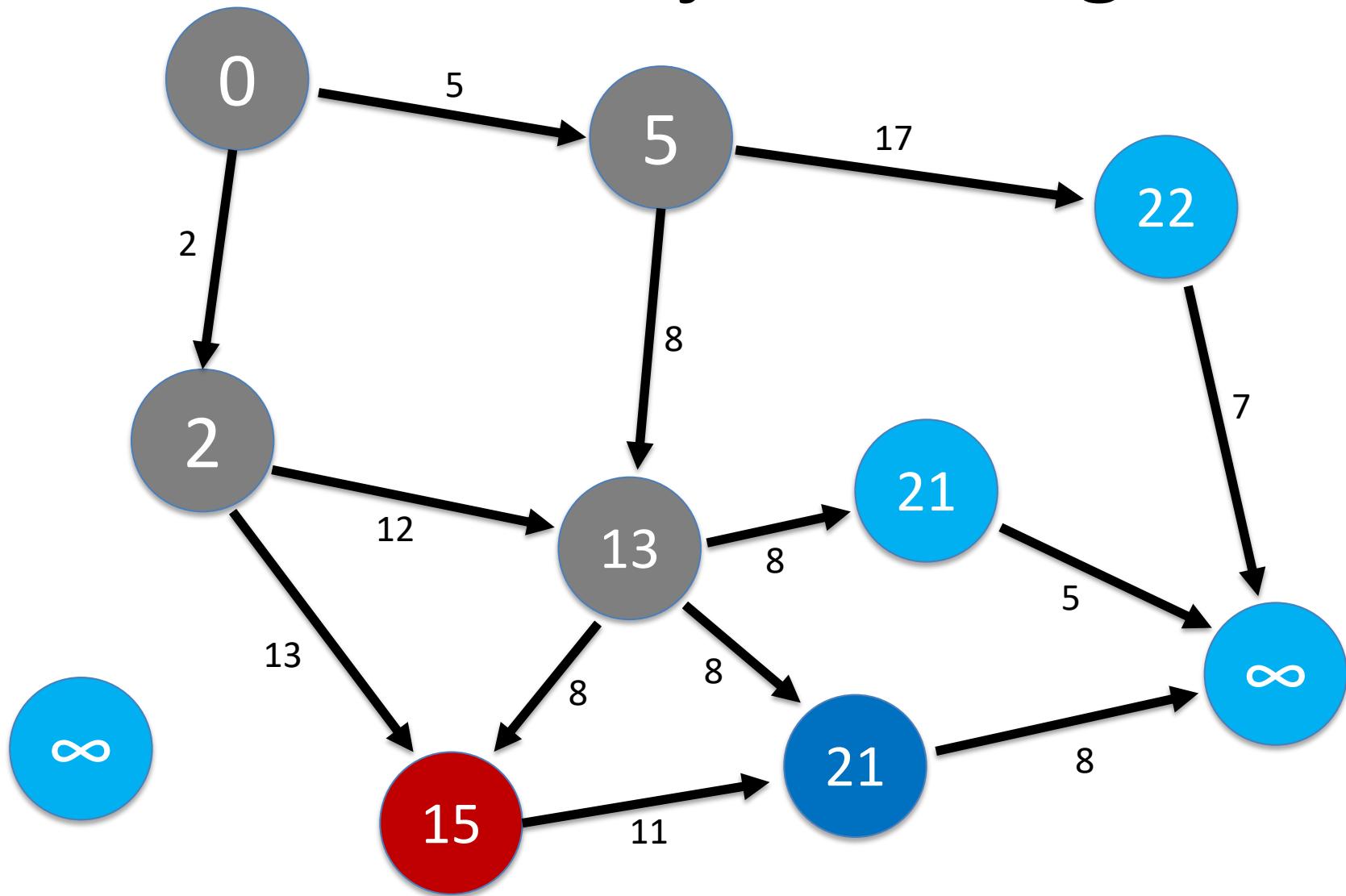
Dijkstra's Algorithm



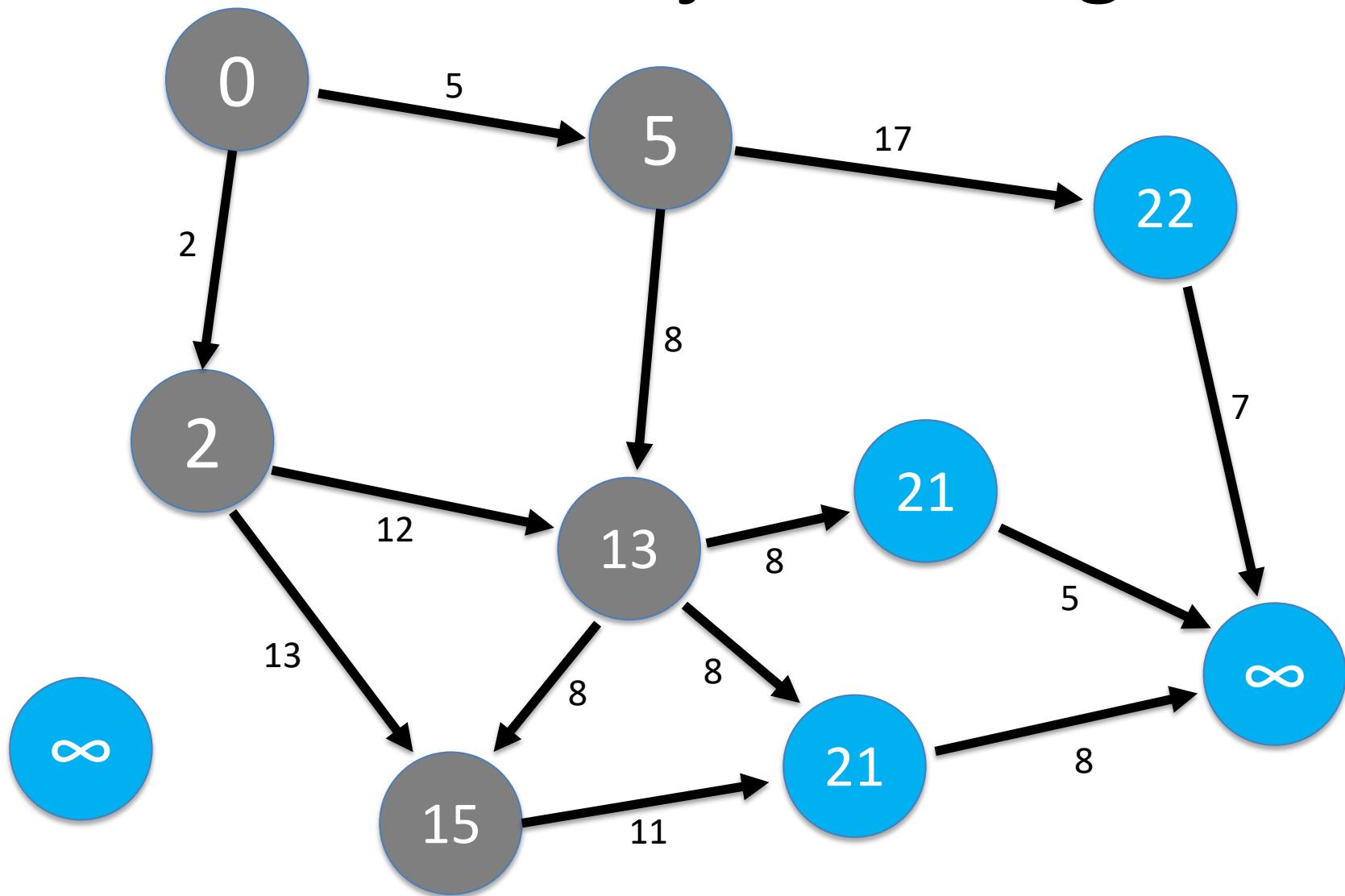
Dijkstra's Algorithm



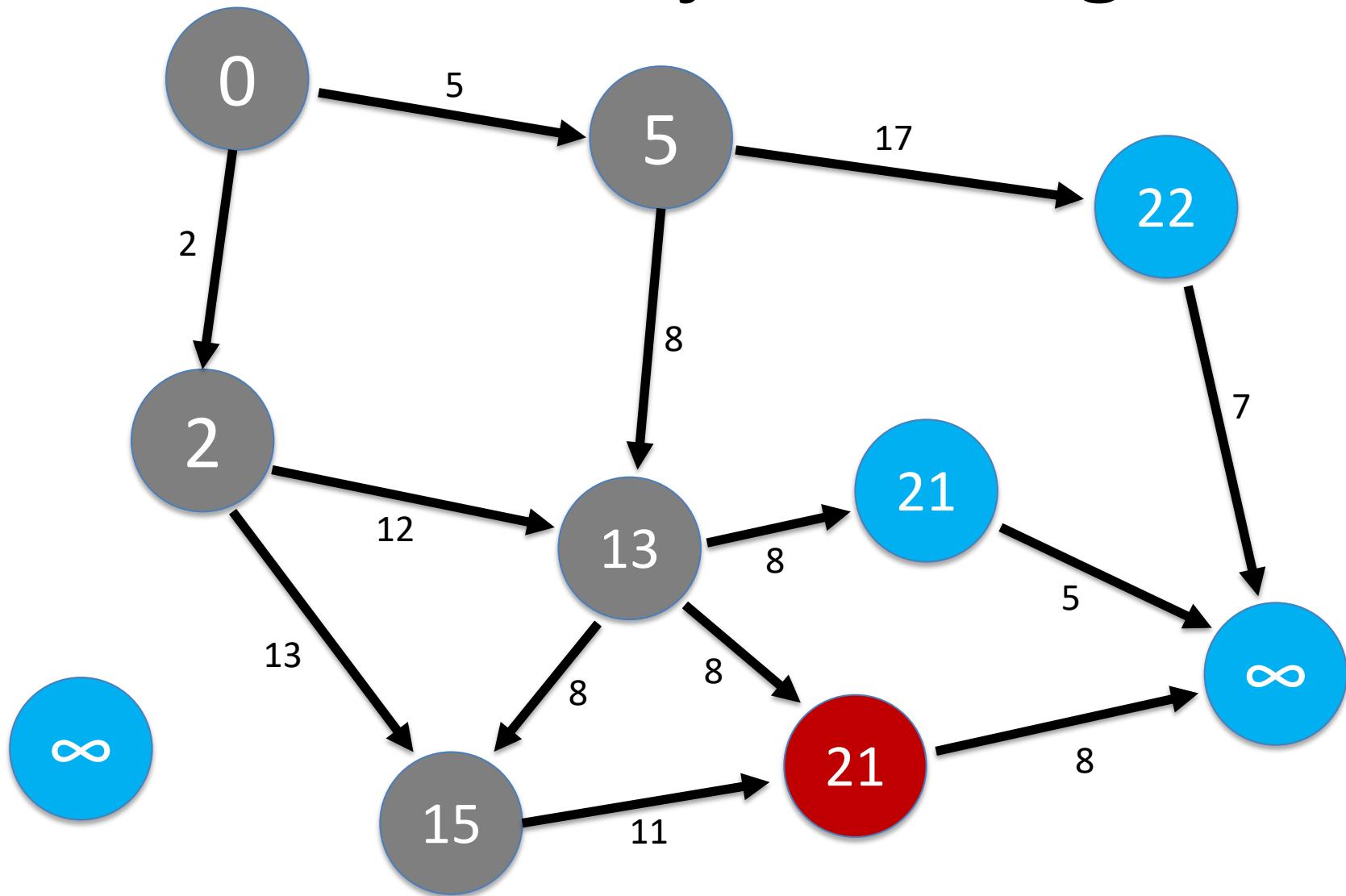
Dijkstra's Algorithm



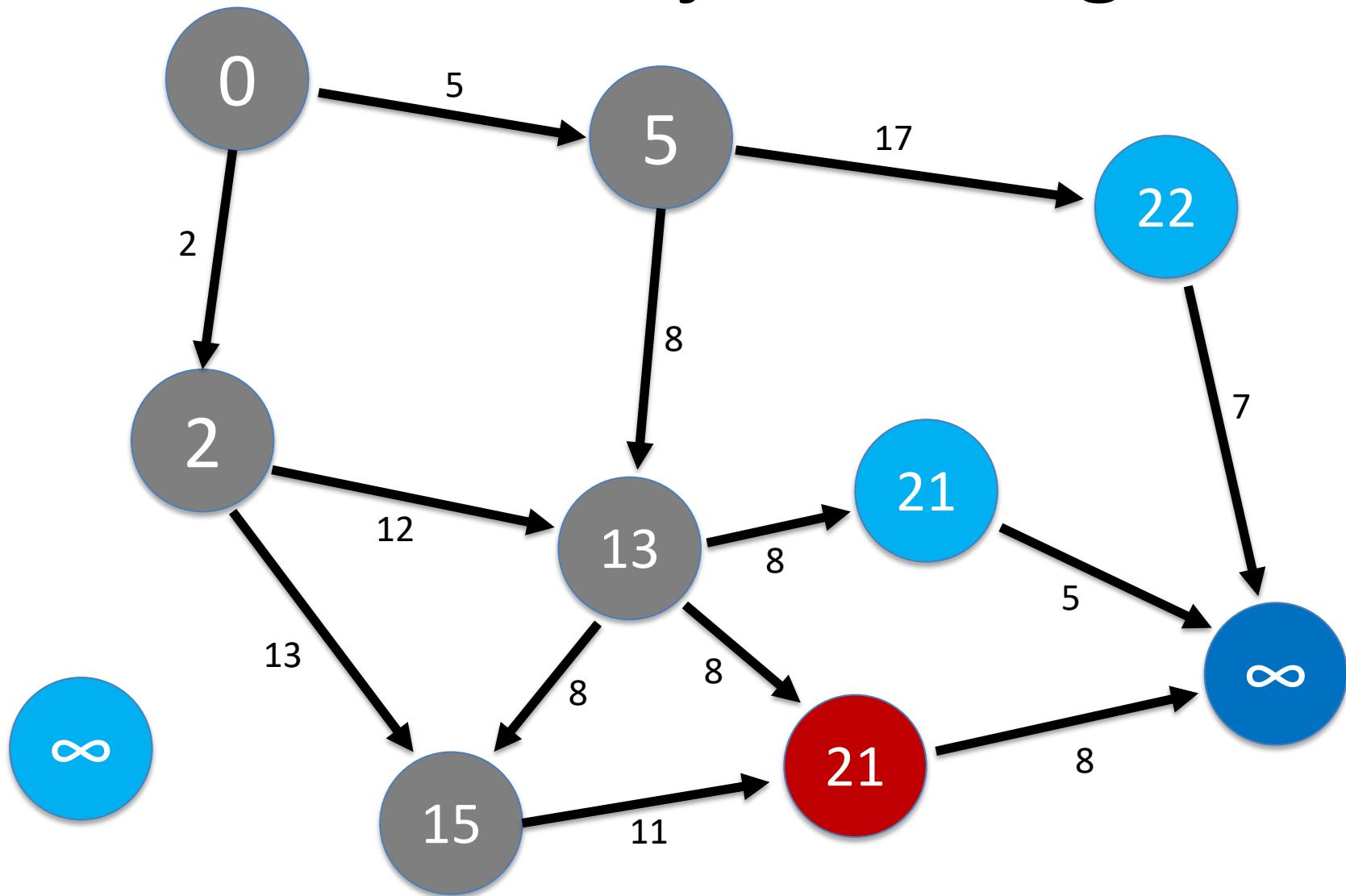
Dijkstra's Algorithm



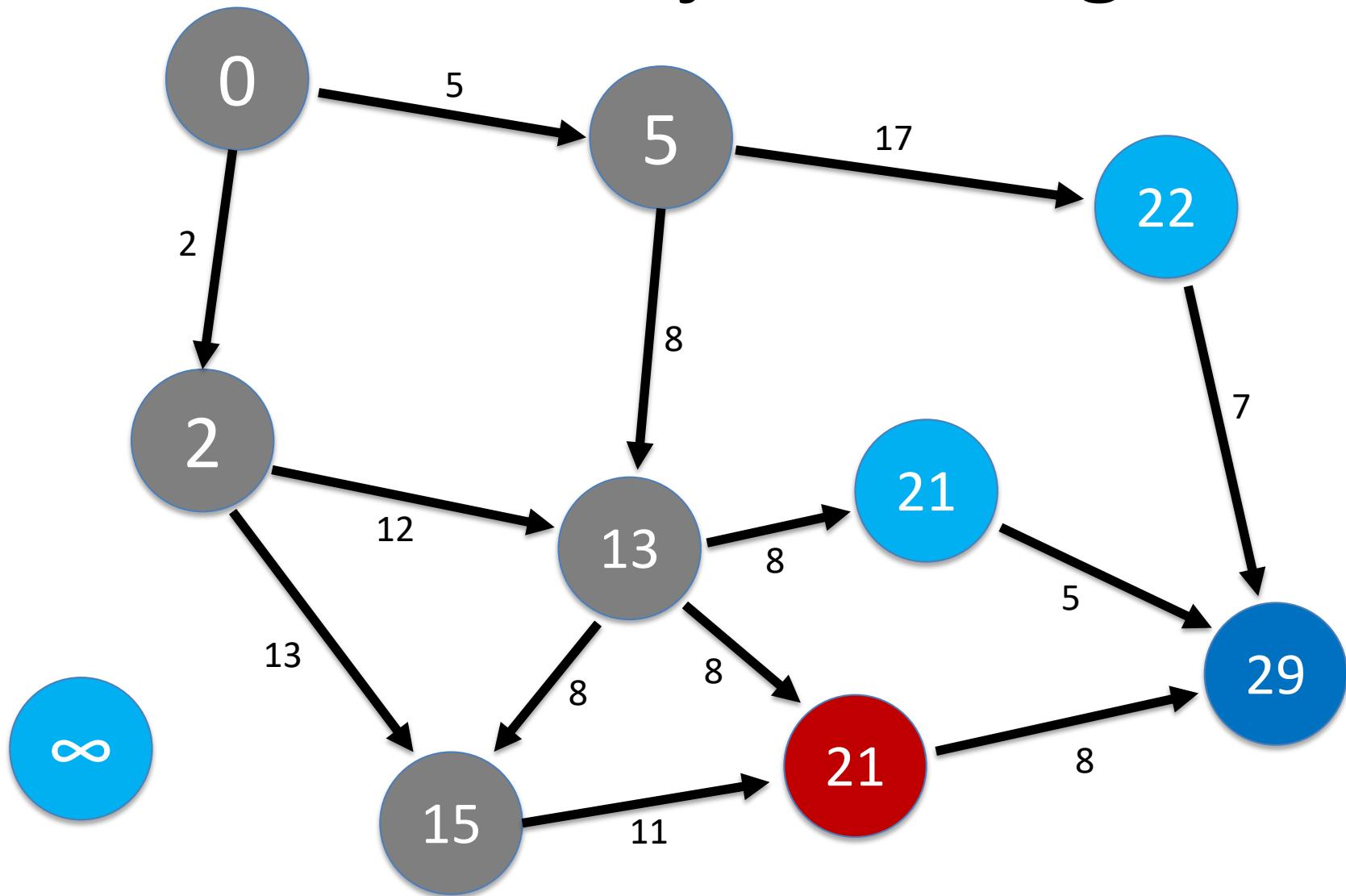
Dijkstra's Algorithm



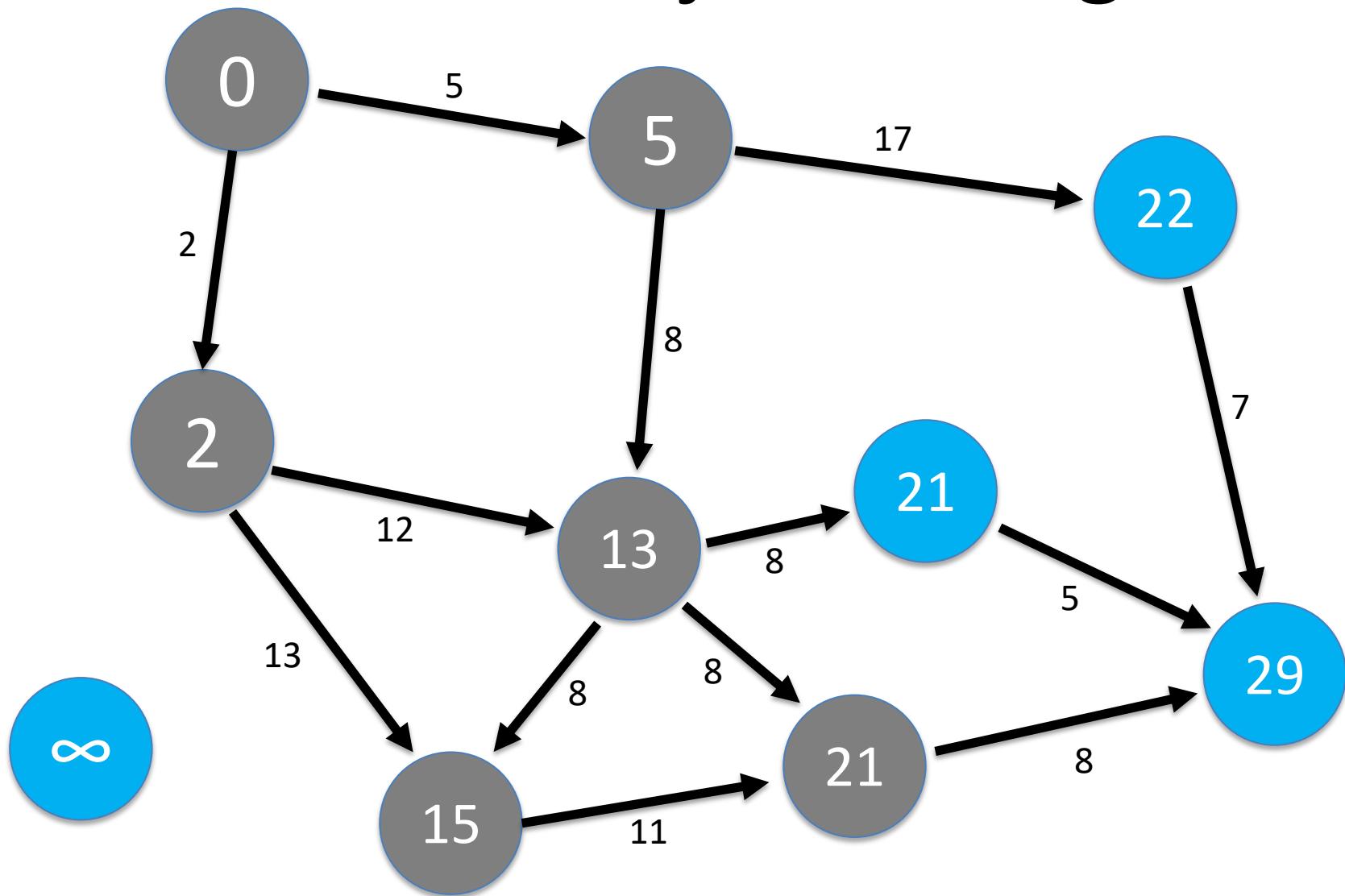
Dijkstra's Algorithm



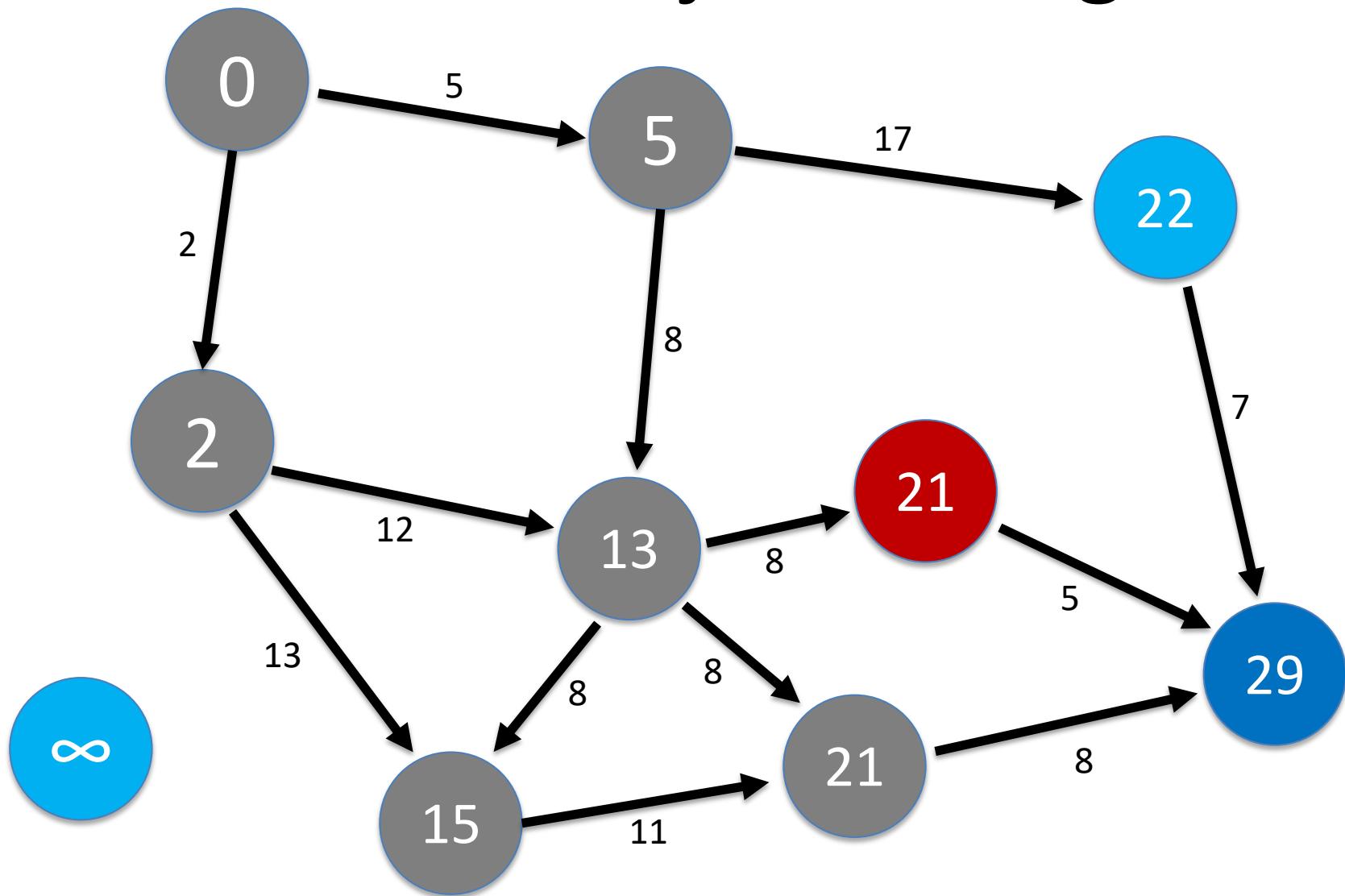
Dijkstra's Algorithm



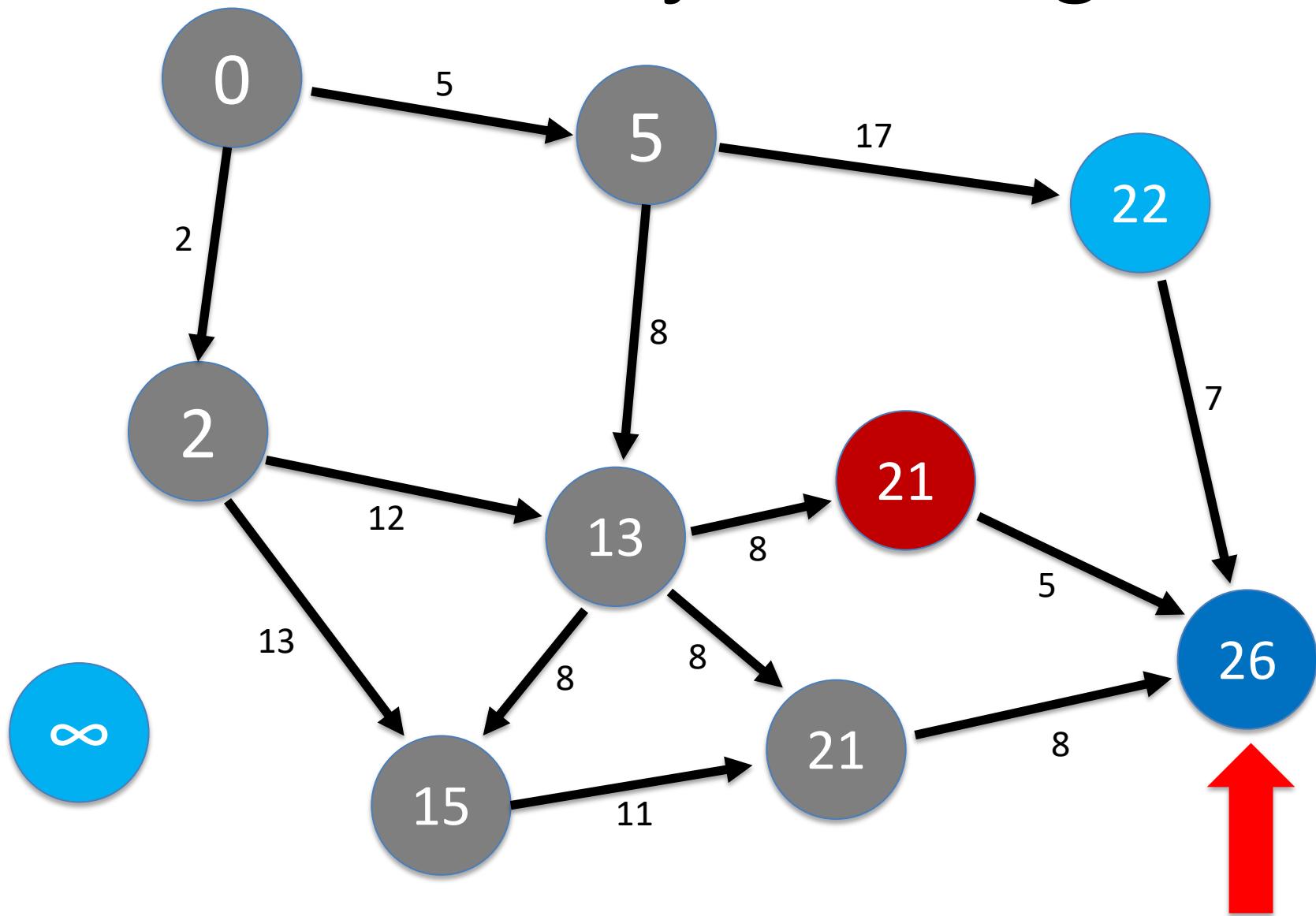
Dijkstra's Algorithm



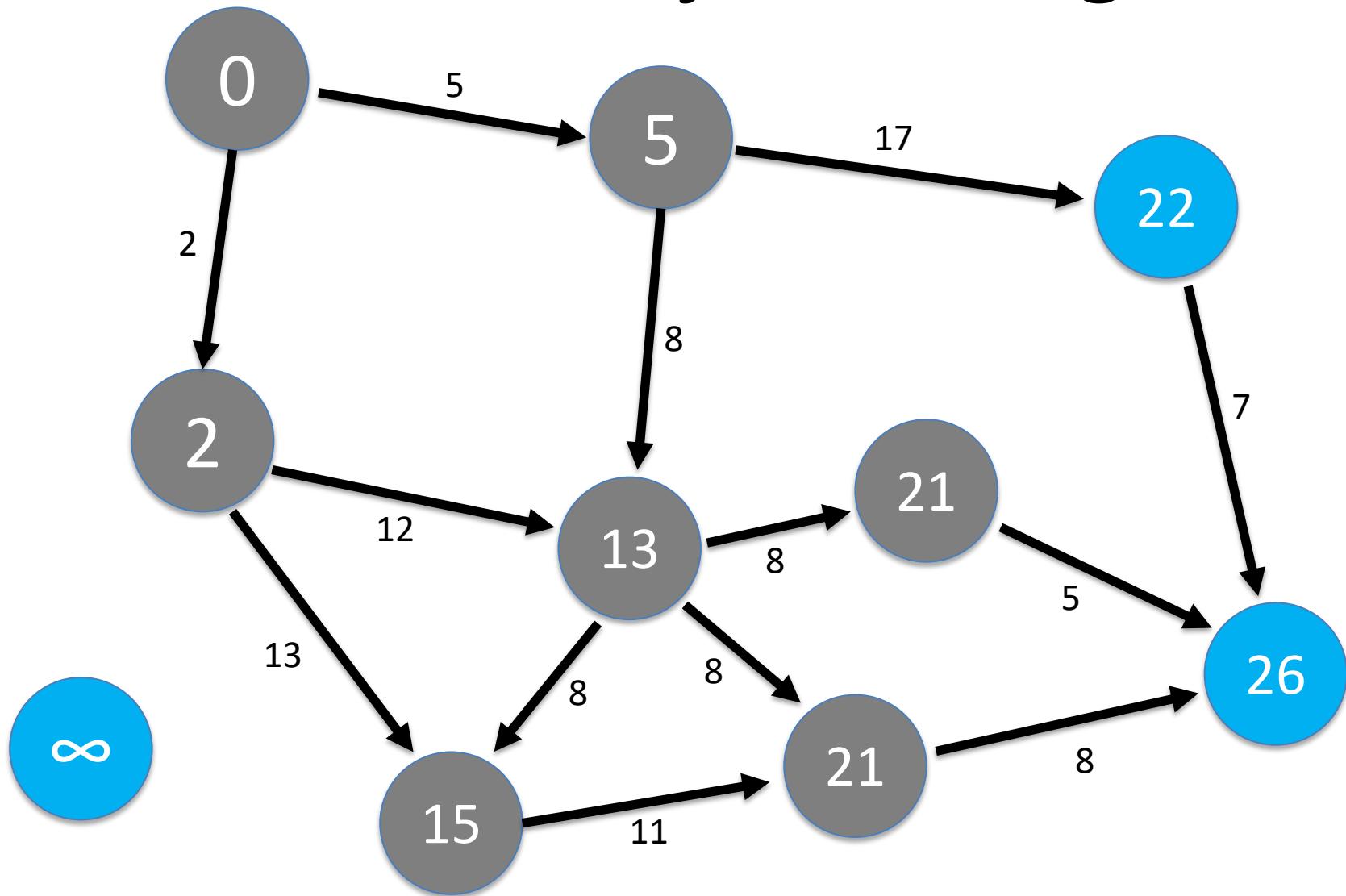
Dijkstra's Algorithm



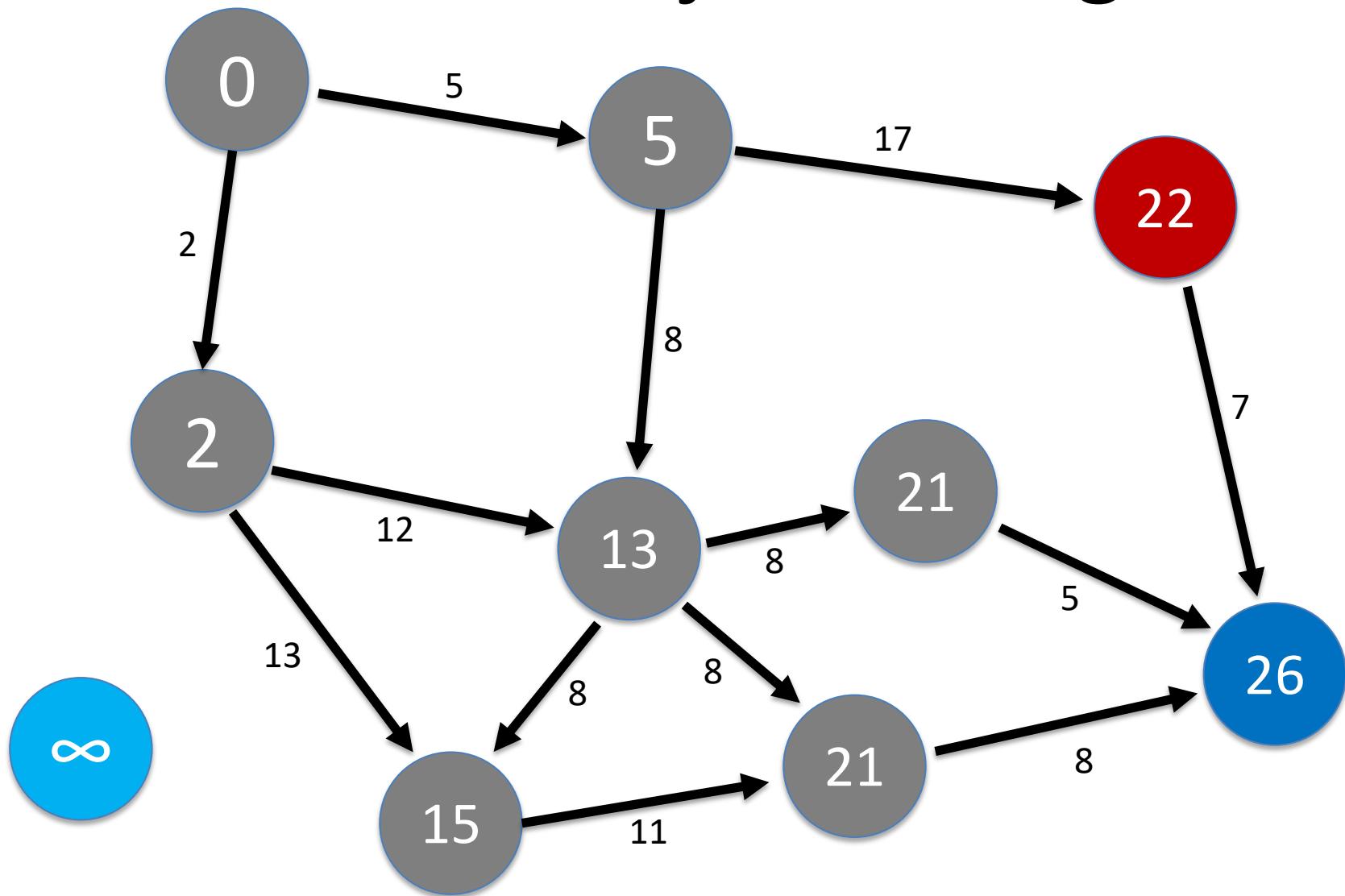
Dijkstra's Algorithm



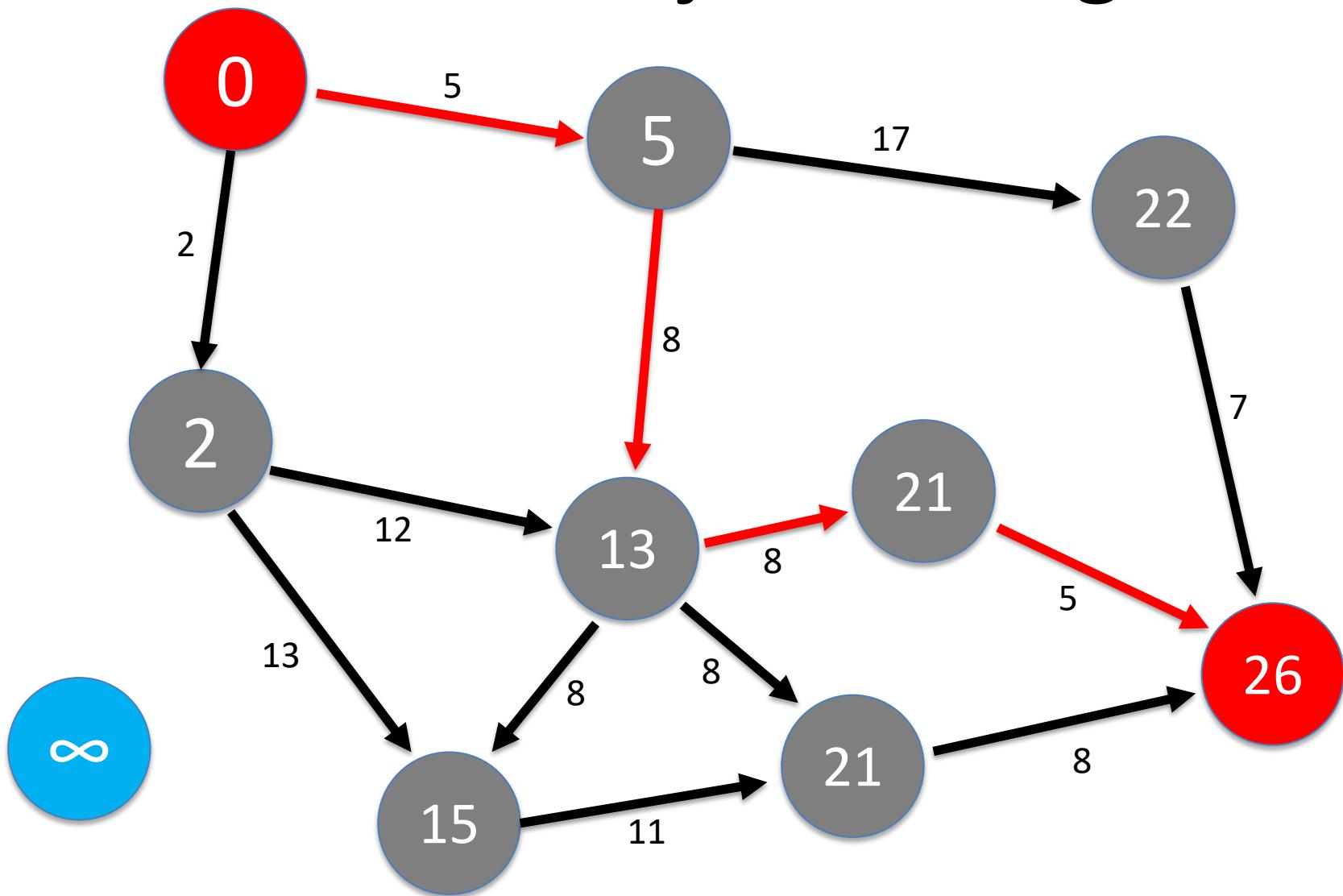
Dijkstra's Algorithm



Dijkstra's Algorithm



Dijkstra's Algorithm



Review

Main Take Aways

- Some important data structures
- Order & Equivalence
- Some general ideas to help with software performance and scale

Some Important Data Structures

- Stacks, Queues, Deques, Priority Queues, Binary Heaps
- Sorting Algorithms: Mergesort, Quicksort, Insertion Sort
- Ordered Maps: BSTs, Balanced BSTs
- Unordered Maps: Hash Tables

Order & Equivalence

- Be mindful of any relational invariants that a data structure should maintain
- Total order – `compareTo`
- Equivalence – `equals`
- If both are in play, they must be *consistent*.

Some General Ideas Relating to Software Performance & Scale

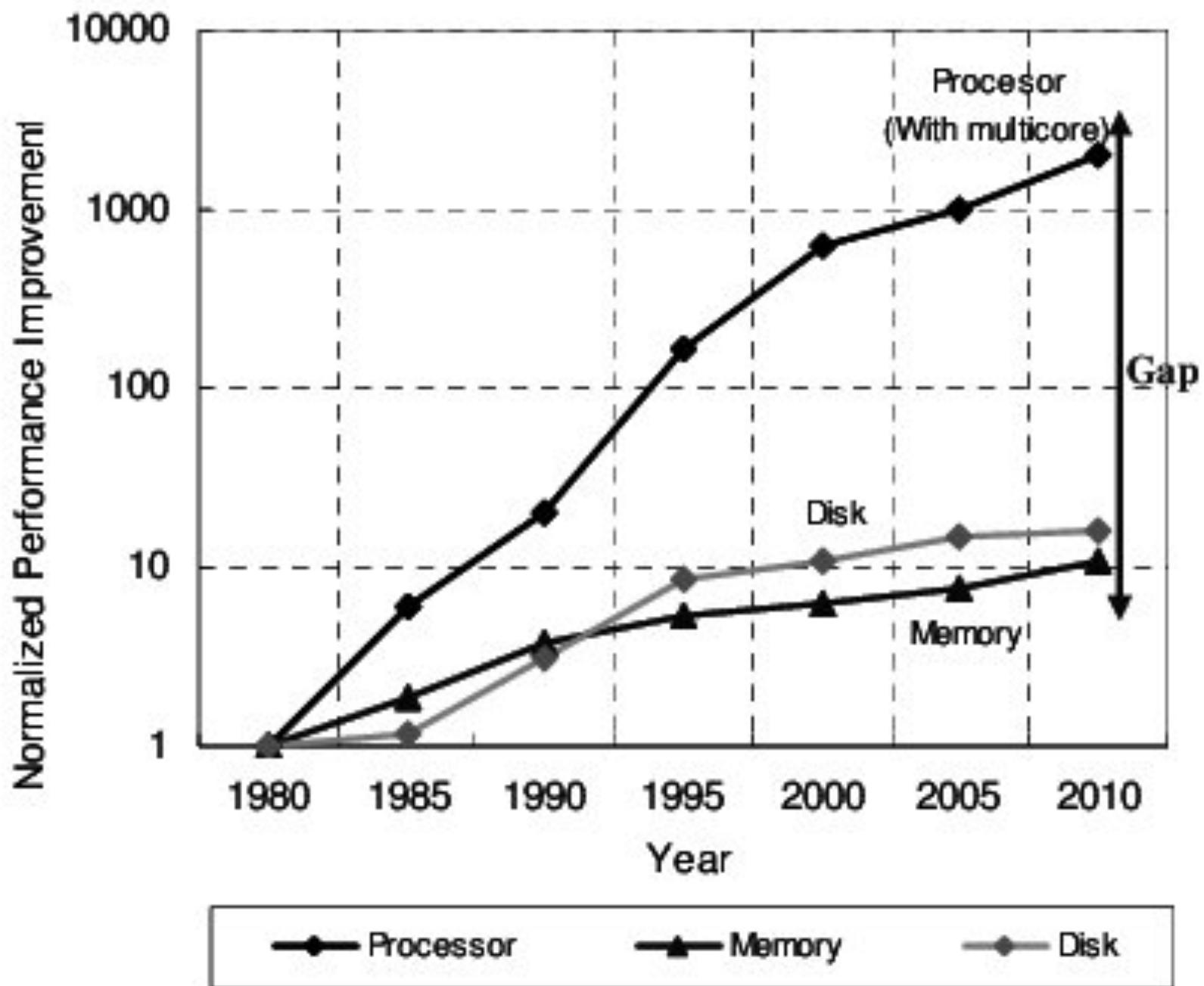
- ADTs promote modularity & support scaling up
 - API
 - Implementations
- What types are required for the expected applications?
- What operations are required?

Some General Ideas Relating to Software Performance & Scale

- Which operations to optimize for the expected applications?
- Given these constraints, which concrete representation types are appropriate?

Linked vs Block Allocation

- Linked allocation is usually more flexible
- Block allocation tends to have good locality



Mutable vs Immutable Types

- Immutable types
 - Reliable – no avoids interference
 - Promote structure sharing
 - Thread safe
 - Can be parallelized
- Mutable types
 - Natural for block allocated data structures

Iteration vs Recursion

- When concrete type is block allocated (e.g., an array), natural to use for-loops or while-loops
- When algorithm is recursive, natural to use recursive implementation
- When a data structure is defined recursively, use recursion on the recursively defined parts

Final Quiz

- Short but covers whole course with an emphasis on material introduced since 2nd quiz
- Drawing diagrams
- See keys to previous final exams

Thank You