UCIrvine | The Henry Samueli School of Engineering

Quantifying Concurrency; Parallel Algorithms

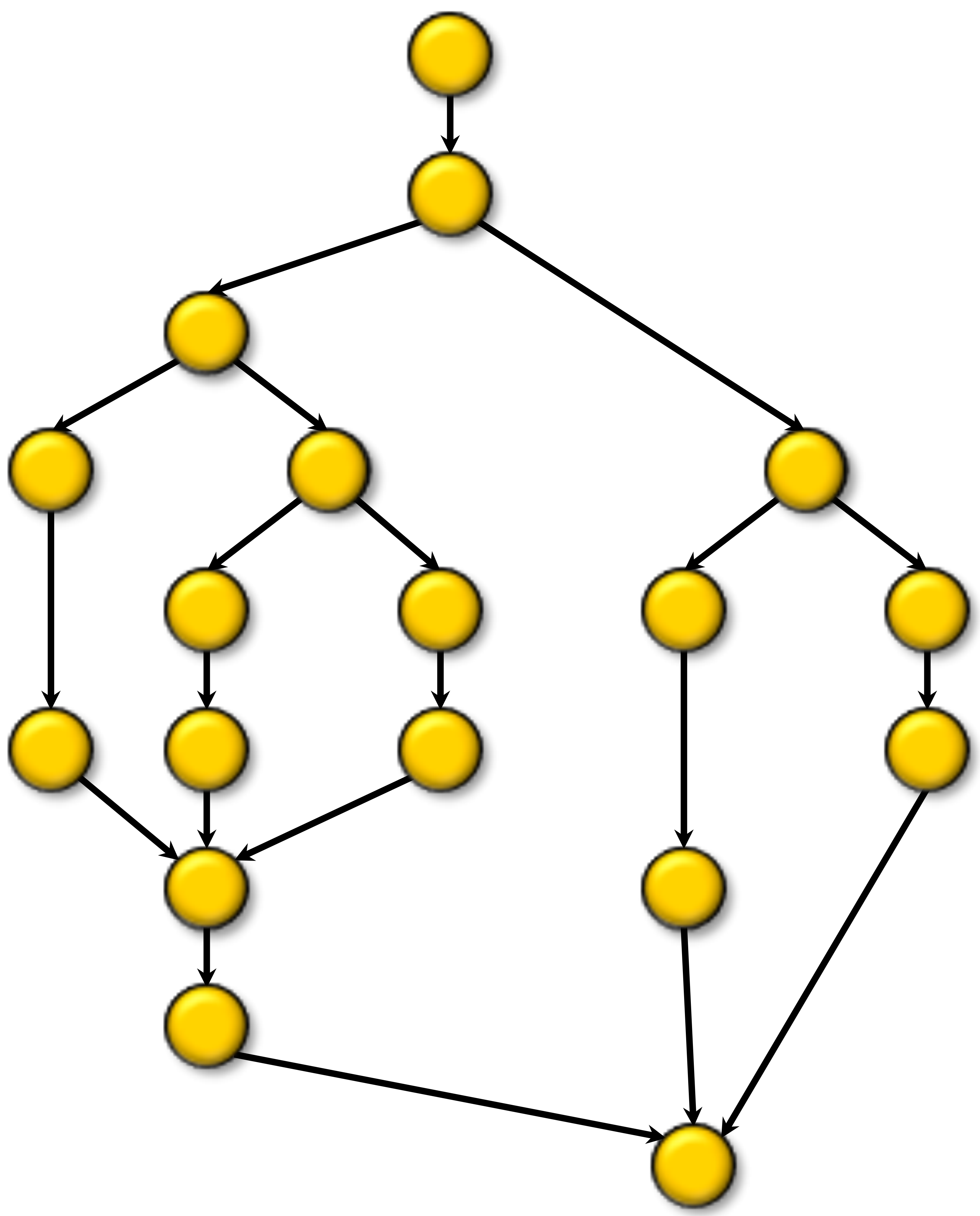
EECS 221: Intro to High-Performance Computing

Aparna Chandramowlishwaran

April 6, 2017

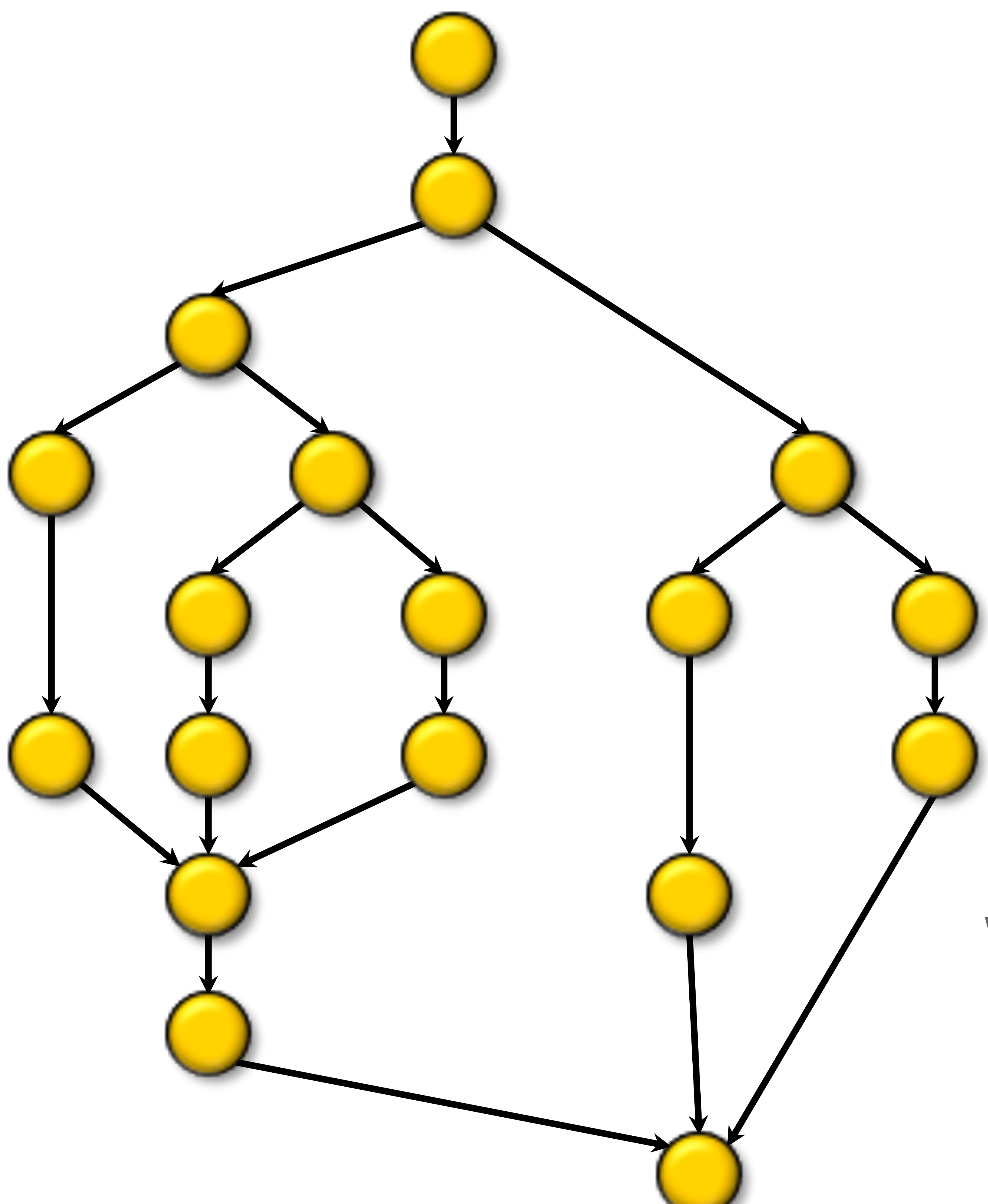
- Quantifying Concurrency:
  DAG, *Work-Depth model, Brent's theorem (1971)*
- First Parallel Algorithm:
  Parallel Mergesort

DAG model of parallel computation



- **Work** = Total number of operations. Could interpret as sequential time.

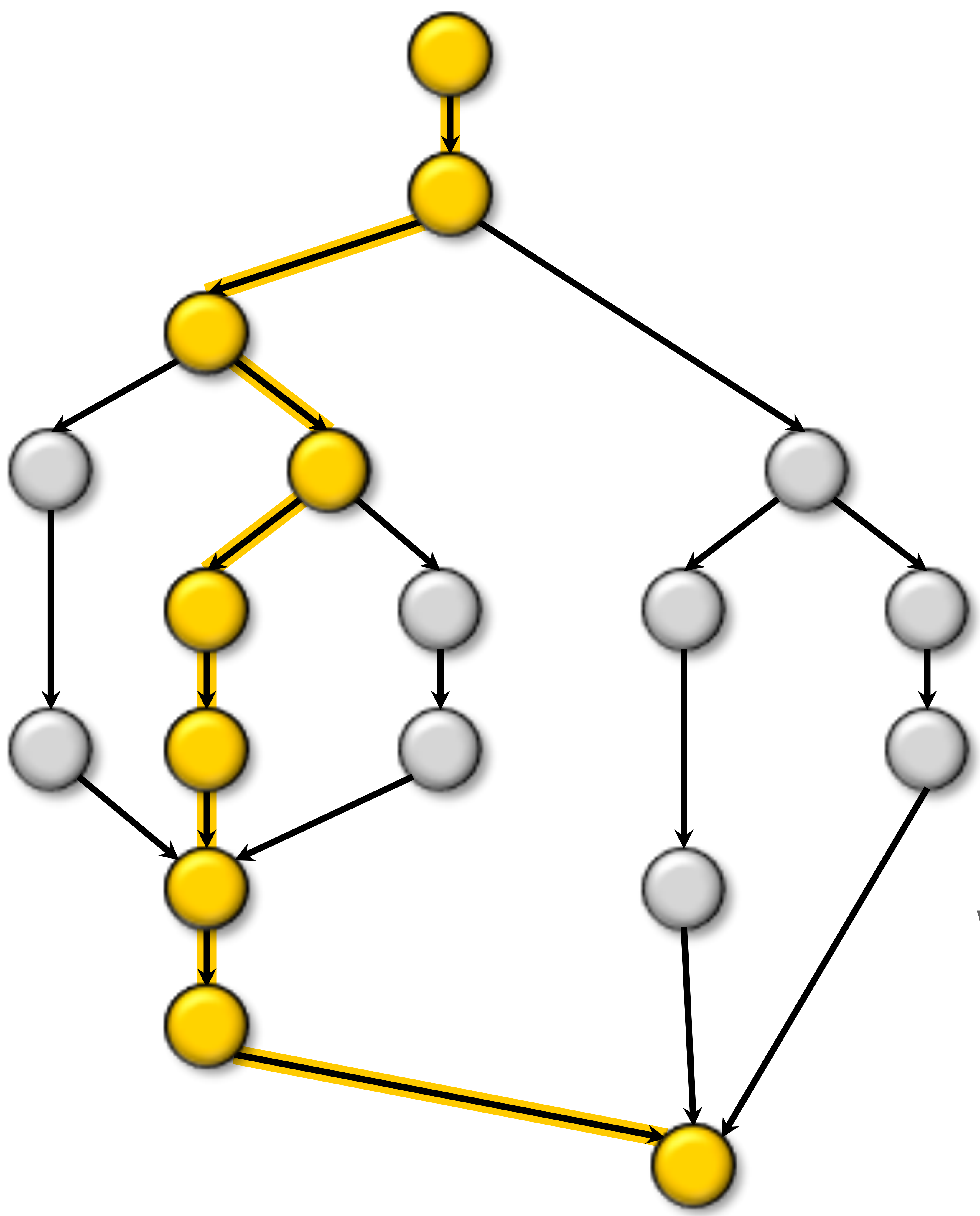
DAG model of parallel computation



W = 18

- **Work** = Total number of operations. Could interpret as sequential time.

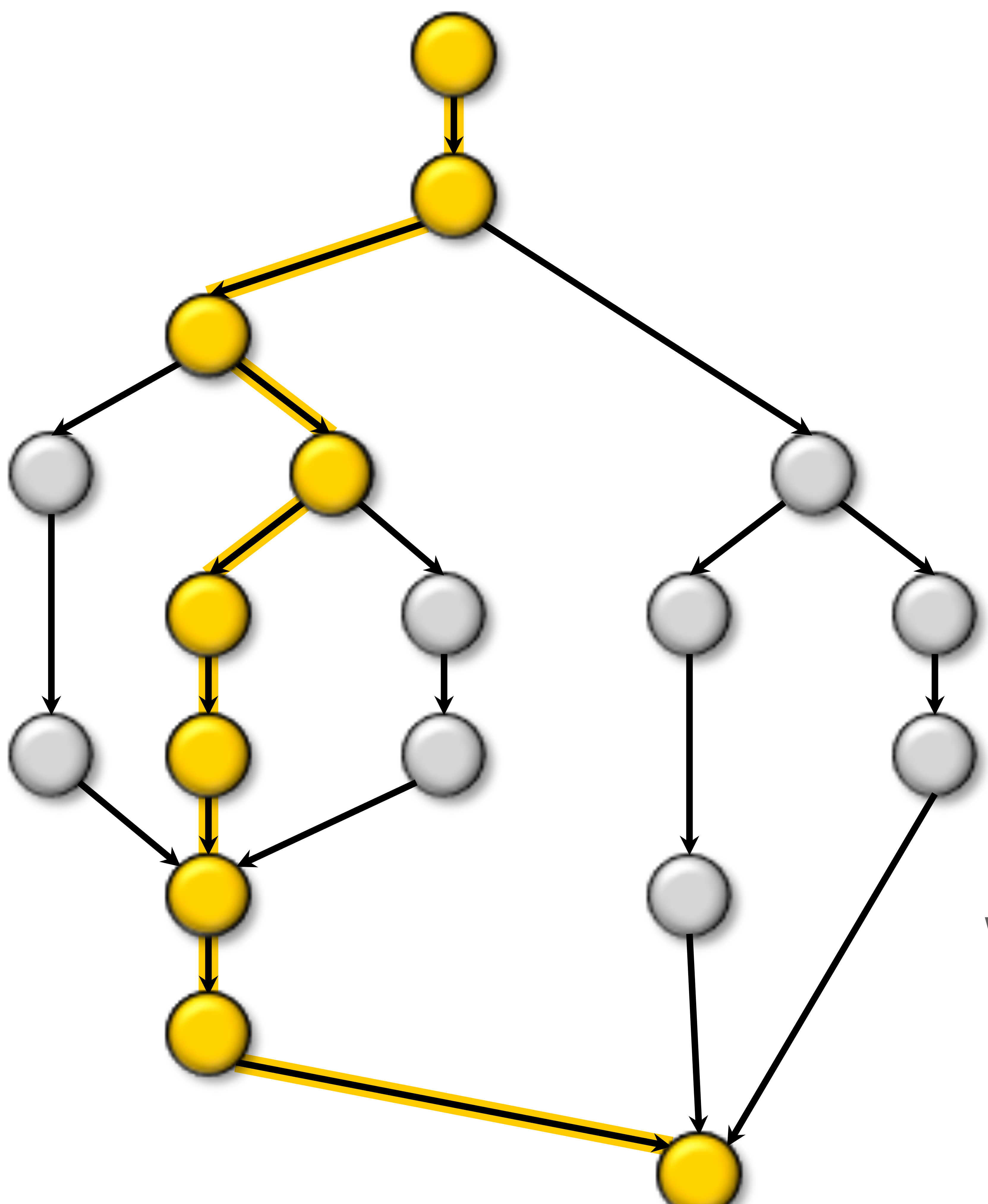
DAG model of parallel computation



W = 18

- **Work** = Total number of operations. Could interpret as sequential time.
- **Depth** = Length of longest dependence chain. Also called critical path.

DAG model of parallel computation



W = 18

D = 9

- **Work** = Total number of operations. Could interpret as sequential time.
- **Depth** = Length of longest dependence chain. Also called critical path.

Relating work, depth, and parallelism

- Work and span laws

$$T_p \geq \frac{W}{p} \qquad T_p \geq T_\infty$$

- Speedup

$$S_p \equiv \frac{W}{T_p} \leq p$$

- Brent's theorem

$$T_p \leq D + \frac{W}{p}$$

- Brent’s theorem proof (whiteboard)

Designing parallel algorithms

- **Work efficiency**: A parallel algorithm is work efficient if it performs asymptotically the same work as the best known sequential algorithm for that problem.

- **Goals**
  - Keep work as low as possible
  - Keep parallelism as high as possible (and hence the depth as low as possible)

- First Parallel Algorithms:
  Parallel mergesort & merge (whiteboard)

Exercises: Compute **W(n)** and **D(n)** for

(1) Summing an array of ***n*** elements

(2) Multiplying ***n*** x ***n*** matrices

(3) Parallel Quicksort