Combinatorial Scheduling:

A way to motivate matrix multiplication and other important concepts

Donna A. Dietz
dietzd@seas.upenn.edu

Wednesday 4th January 2012
4:35pm
1077-F1-274

MAA Session
on
Innovative and Effective Ways
to Teach Linear Algebra

JMM Meeting
Boston

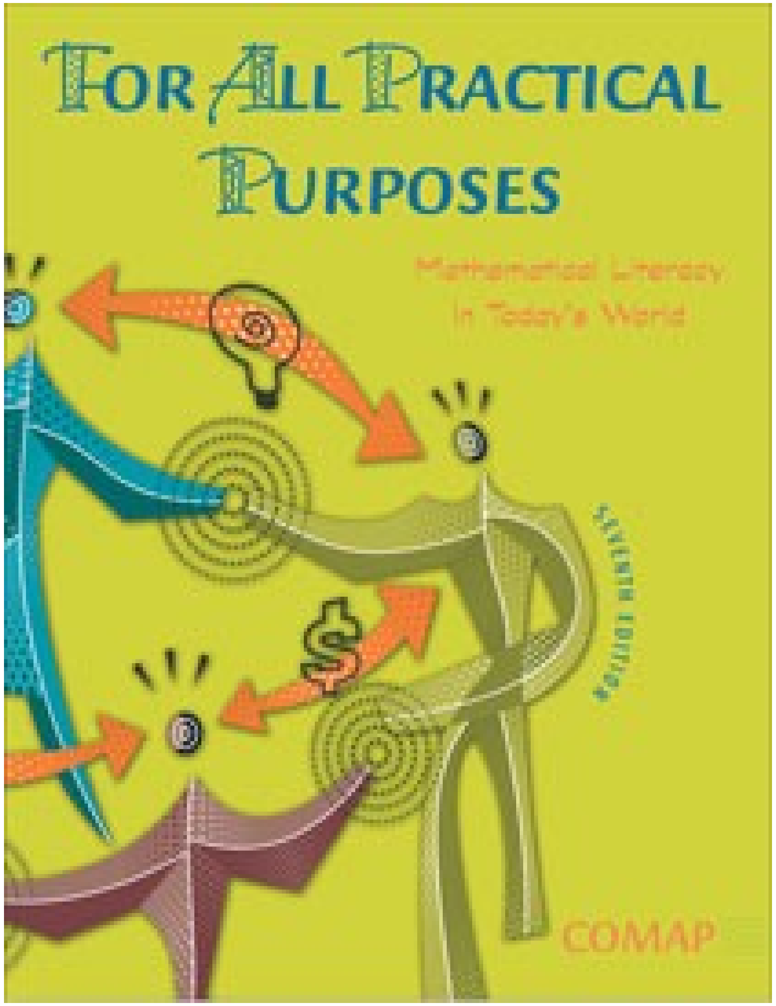
MCIT Program at Penn:

- Master's of Computer Information Technology
- 10 courses: 6 core, 4 electives
- Strong students, various backgrounds
- About 80 students
- 3 Lecturers
  - Programming (David Matuszek, roughly 10th year.)
  - Mathematics (I am in my third year.)
  - Architecture (Chris Murphy, second year.)

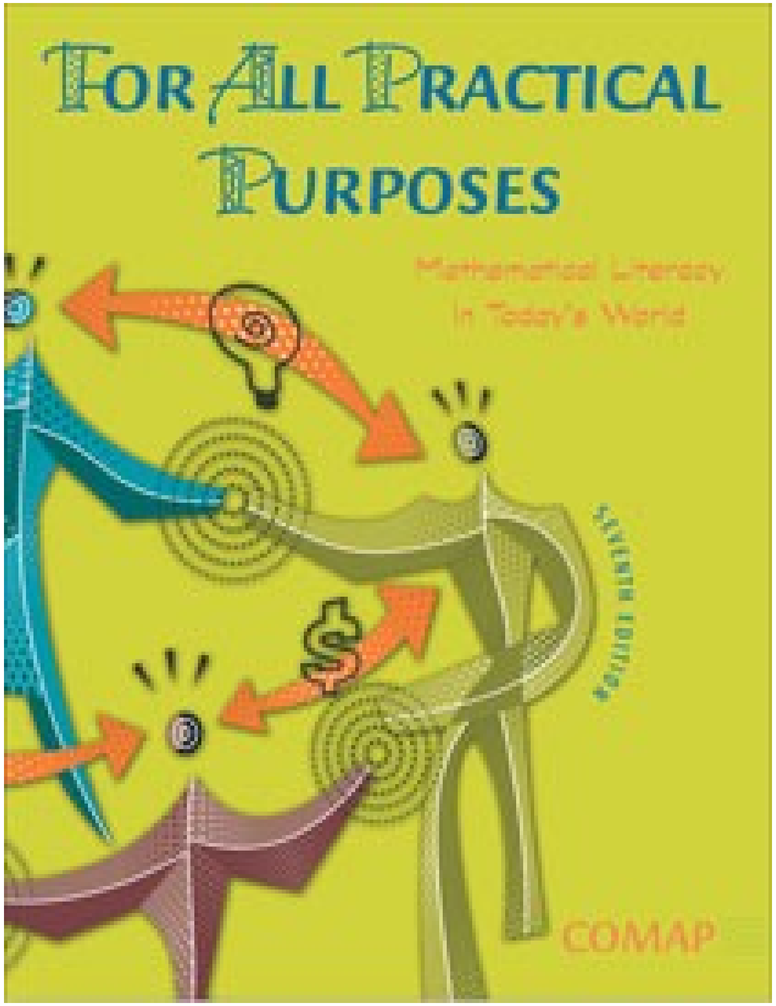
Problem:

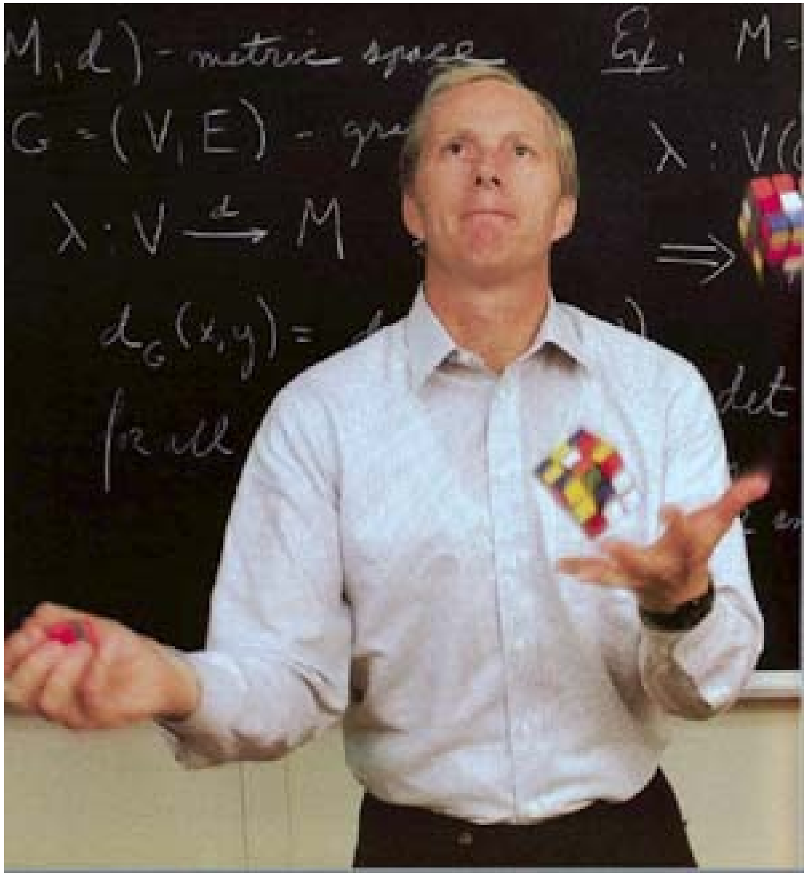
Given a list of tasks,
their completion times,
and their prerequisites,
come up with a schedule for them
on *n* processors.

Problem Source:



Problem Source:

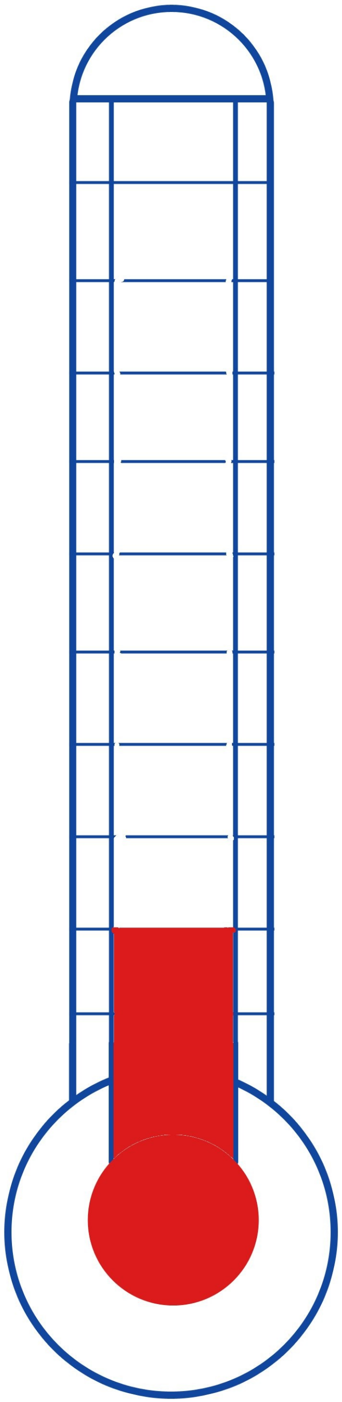




Ronald Graham

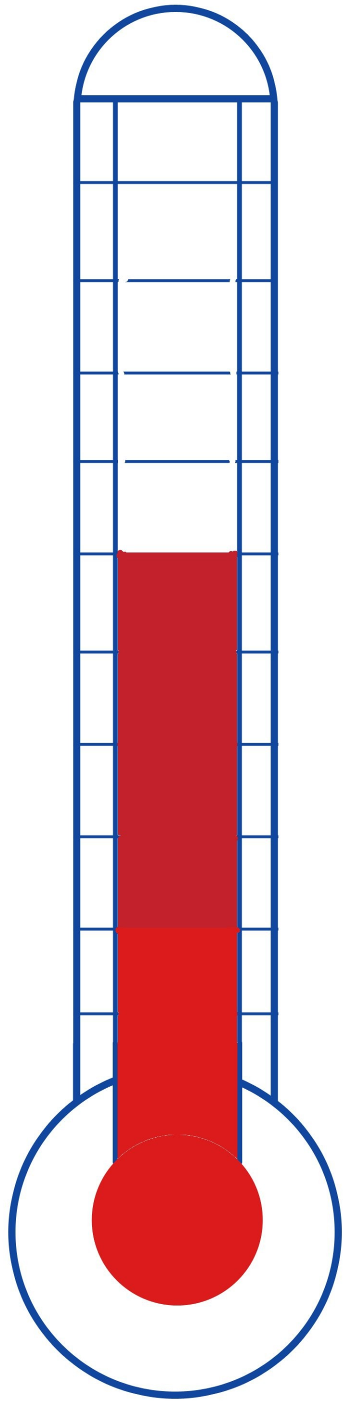
UCSD

Difficulty Level of Problem:



By-hand calculations:
Appropriate for General
Education ("Math
Appreciation") Courses

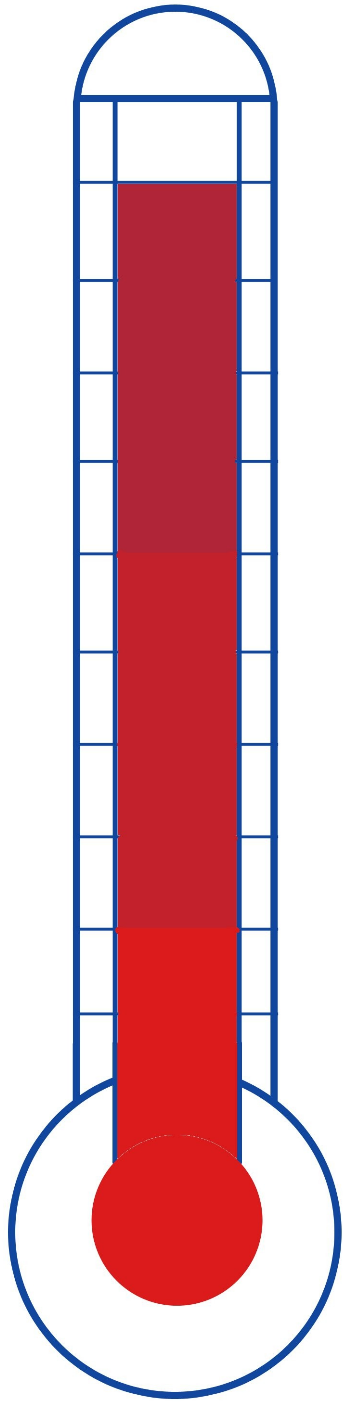
Difficulty Level of Problem:



Matrix multiplication to find paths in graphs. (Undergraduate Discrete, Linear Algebra courses.)

By-hand calculations: Appropriate for General Education ("Math Appreciation") Courses

Difficulty Level of Problem:

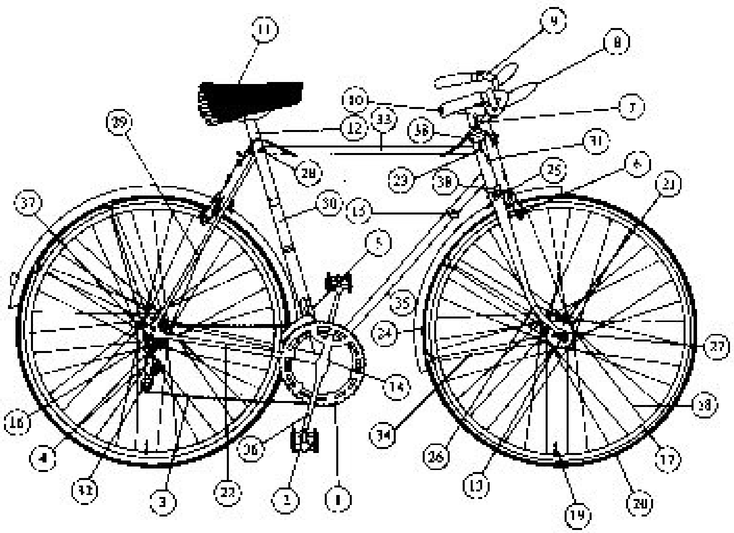


Programming Project for advanced first semester course or second semester course.

Matrix multiplication to find paths in graphs. (Undergraduate Discrete, Linear Algebra courses.)

By-hand calculations: Appropriate for General Education ("Math Appreciation") Courses

Example Problem:
(By Hand)



FP —Frame preparation which includes installation of the front fork and fenders.

FW—Mounting and aligning front wheel.

BW—Mounting and aligning back wheel.

DE   Attaching the derailleur to the frame.

GC—Installing the gear cluster.

CW—Attaching the chain wheel to the crank.

CR —Attaching the crank and chain wheel to the frame.

RP —Mounting right pedal and toe clip.

LP —Mounting left pedal and toe clip.

FA —Final attachments which includes mounting and adjusting handlebars, seat, brakes, etc.

INPUT to algorithm:
1) Precedence Constraints List with Tasks' Times
2) Number of Processors (used in second half)

```
FP 7
FW 7 FP
BW 7 GC DE FP
DE 2
GC 3 DE
CW 2 DE
CR 2 CW
RP 8 CR CW GC
LP 8 CR CW GC
FA 18 FP FW BW GC DE
```

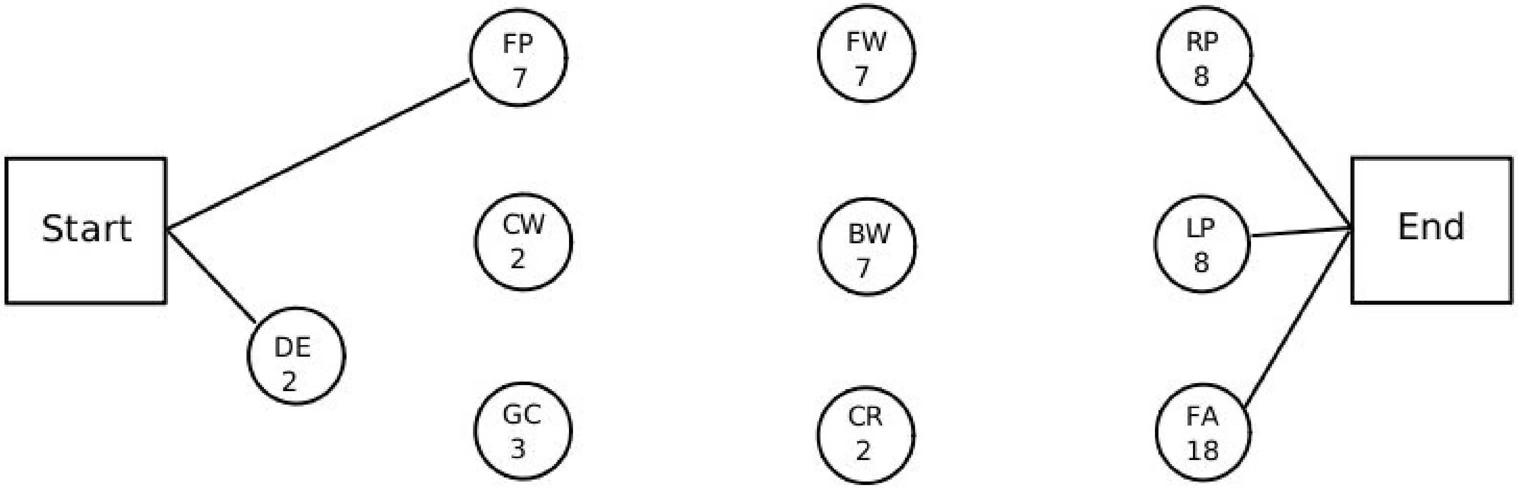
INPUT to algorithm:
1) Precedence Constraints List with Tasks' Times
2) Number of Processors (used in second half)

```
FP 7
FW 7 FP
BW 7 GC DE FP
DE 2
GC 3 DE
CW 2 DE
CR 2 CW
RP 8 CR CW GC
LP 8 CR CW GC
FA 18 FP FW BW GC DE
```

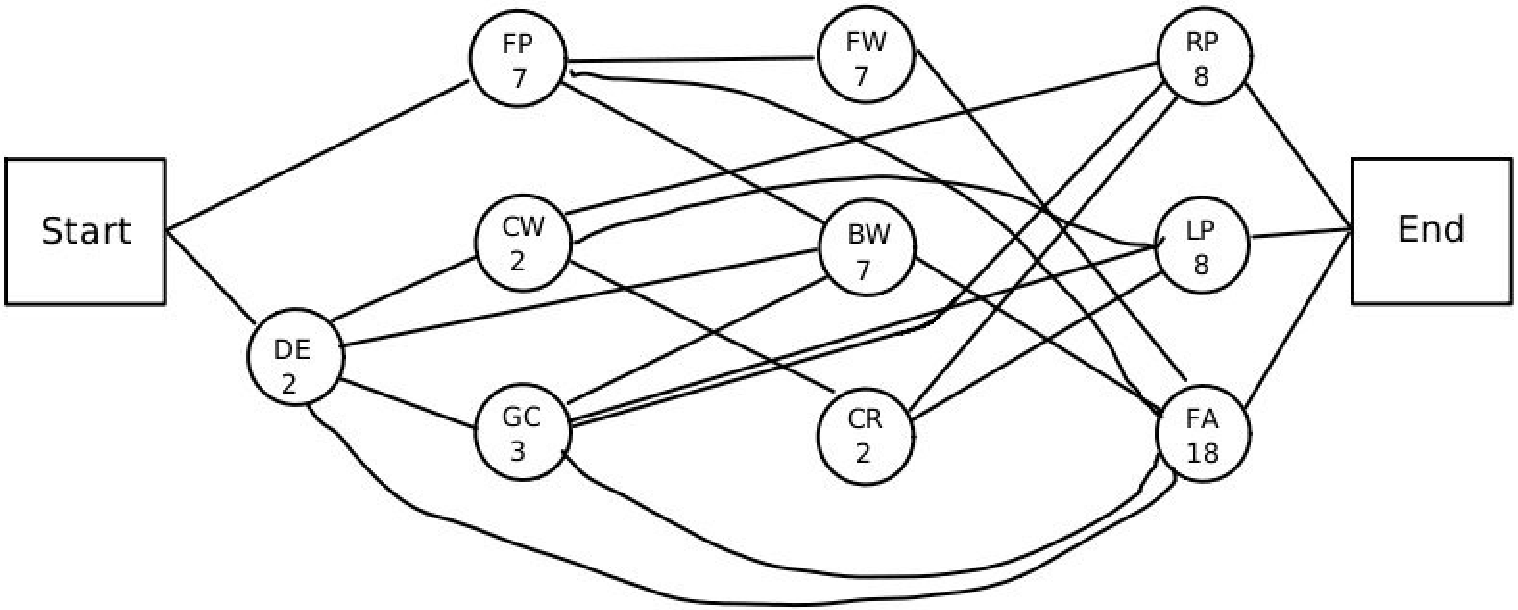
Frame Preparation takes 7 minutes

Gear Cluster takes 3 minutes. Cannot start until derailleur is done.

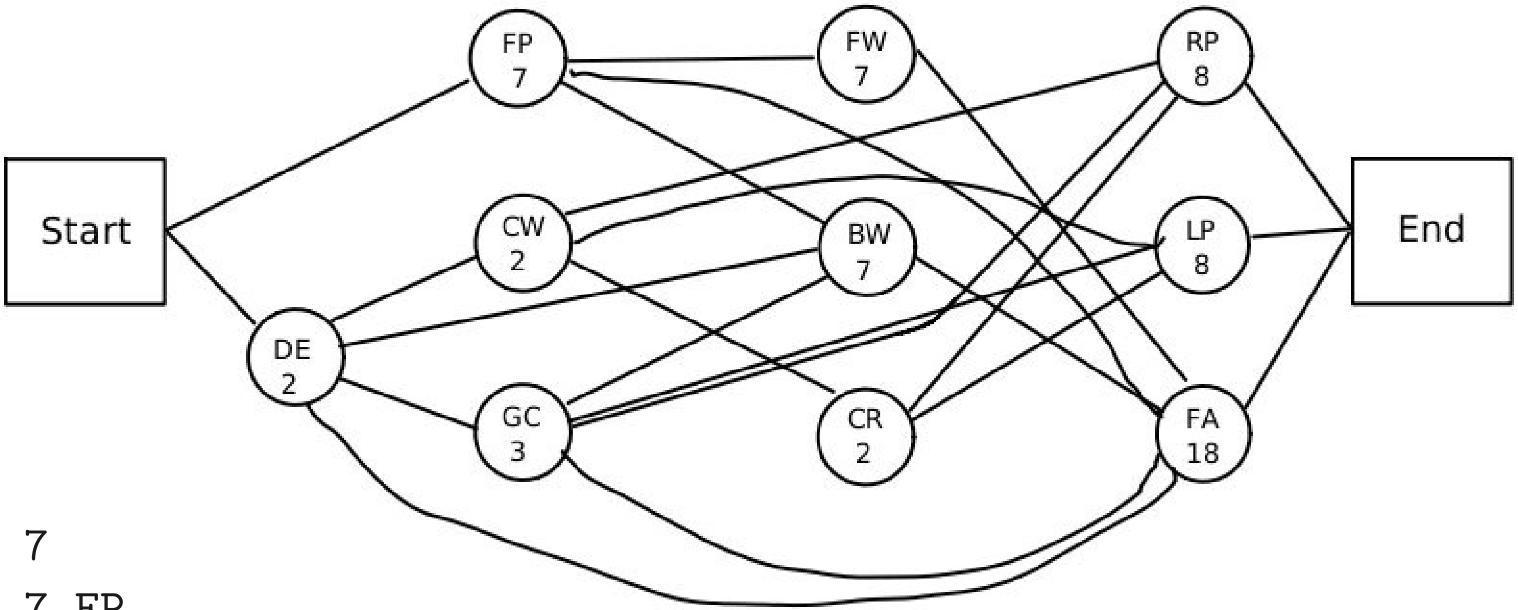
Final Attachements take 18 minutes and cannot start until Frame Prep, both wheels, Gear Cluster and Derailleur are done.



To make algorithm easier to implement, add Start and End tasks.
Connect comparable items to indicate prerequisite relationship.



WHAT A MESS!!!



FP 7

FW 7 FP

BW 7 GC DE FP

DE 2

GC 3 DE

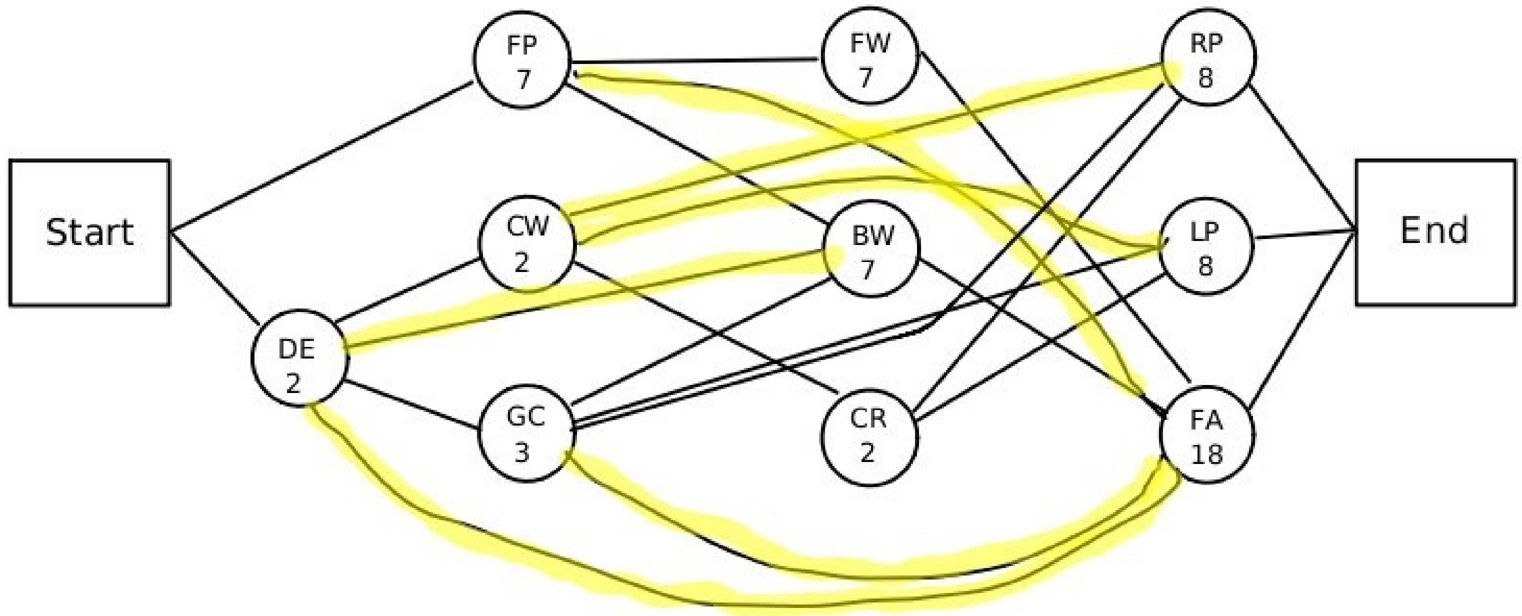
CW 2 DE

CR 2 CW

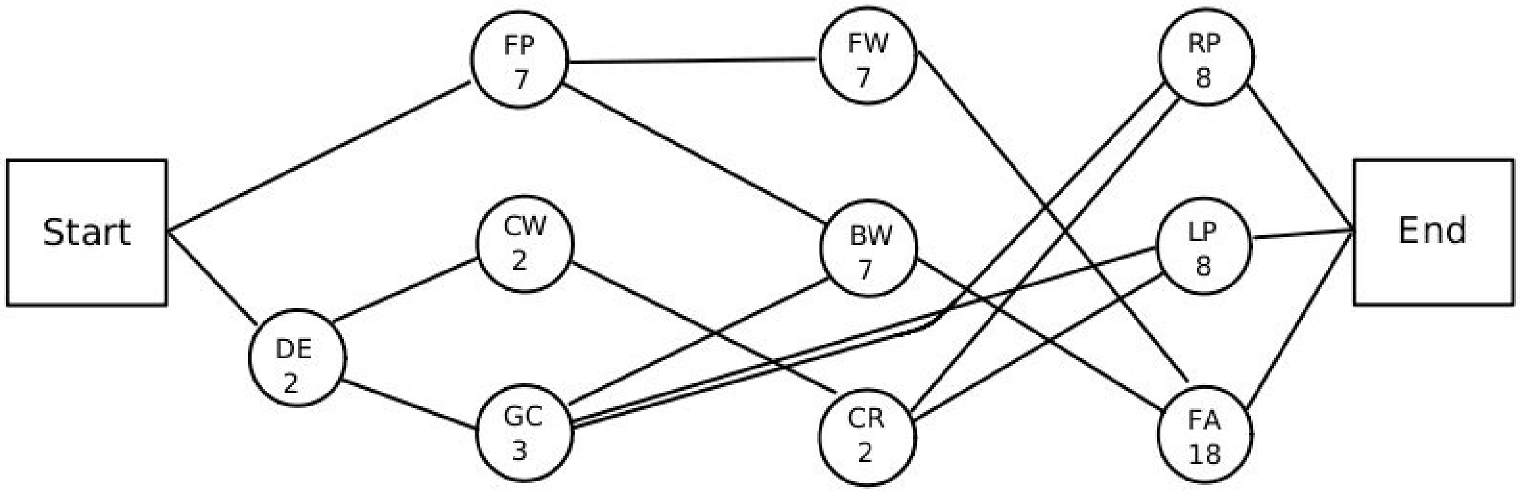
RP 8 CR CW GC

LP 8 CR CW GC

FA 18 FP FW BW GC DE

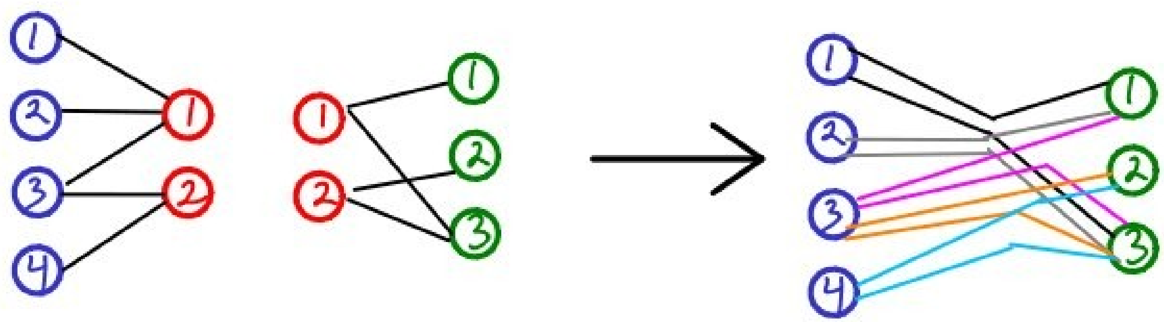


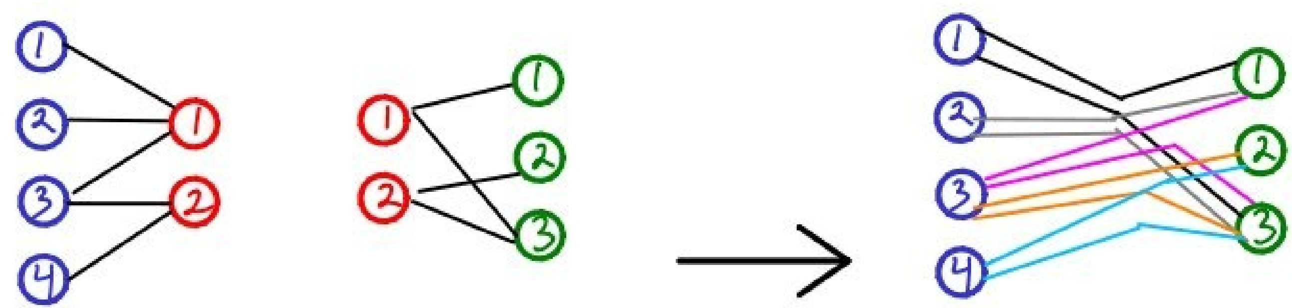
We need to remove the redundancies. (This is a white lie, but the students don't seem to mind. It improves the algorithm's efficiency, and they can see that when they compute these by hand.)



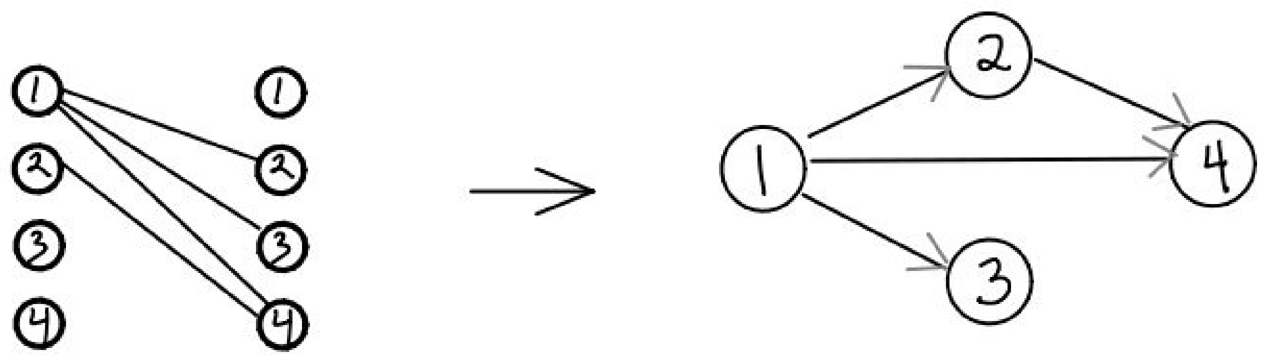
AHHHH! Much Better!

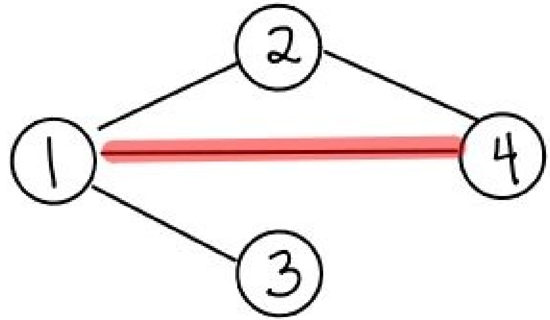
For Linear Algebra: Matrix Multiplication



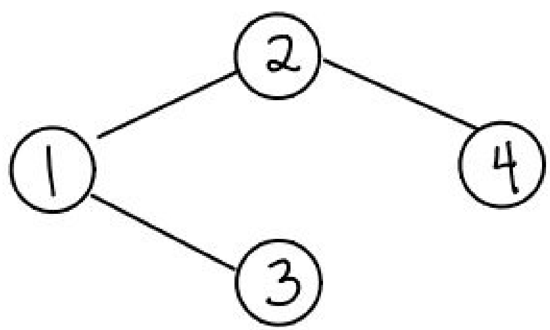


$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 2 \\ 0 & 1 & 1 \end{bmatrix}$$

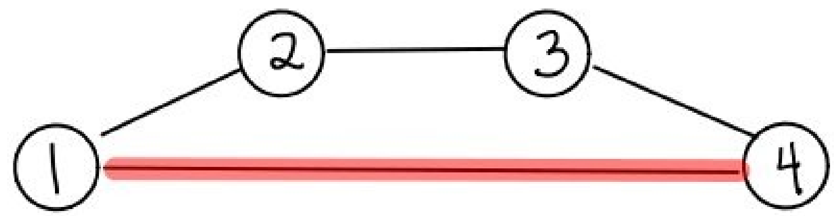

1
2
3
4
1
2
3
4
2
1
4
3



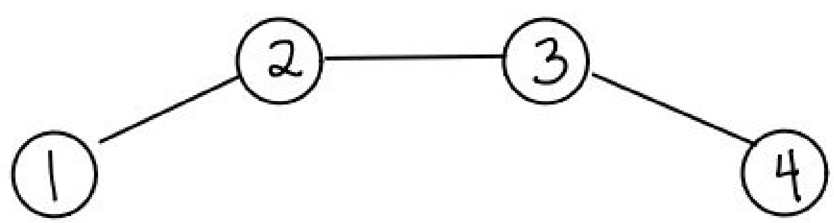
$$M = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad M^2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

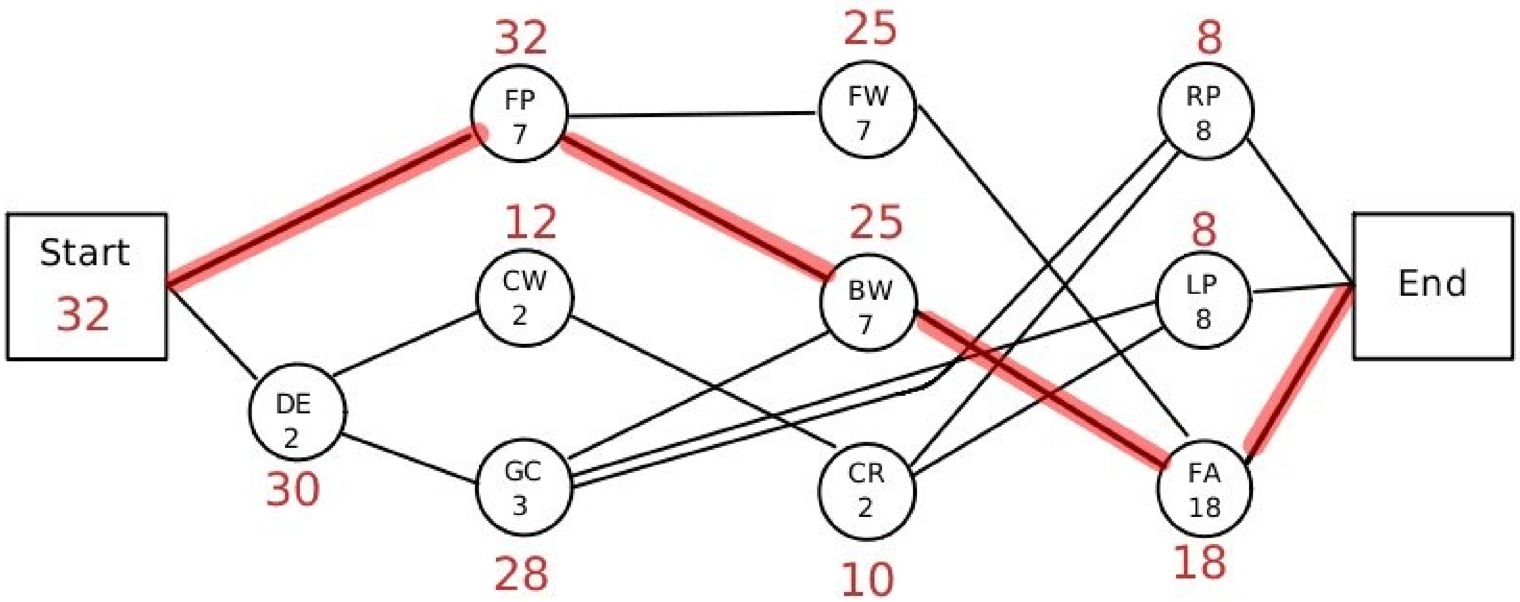


Repeat with new M, n times (where M is an nxn matrix).

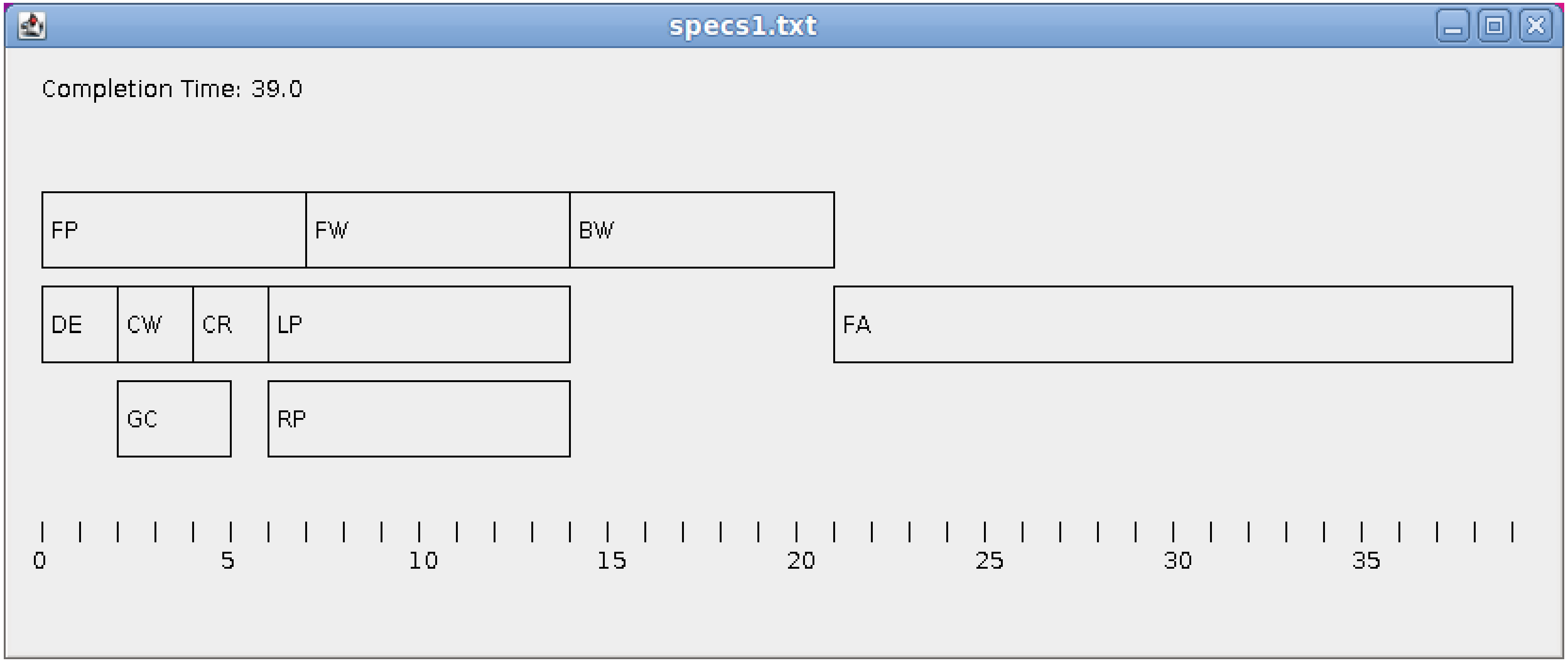


$$M = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdots \; M^3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$





Assign priorities to the tasks using the critical path (CP) length from that task to the End task.
(Overall CP is in red.)



We now make use of the number of processors.
Assign tasks to processors in order of priority, with the caveat that the task cannot run until its prerequisites are finished.

This is reasonably easy to do by hand, but it takes a small amount of practice. However, due to wait periods, non-integer task times, etc, programming this is a great thinking exercise!

CP Scheduling Surprises

- NP Complete
- Paradoxes
  - More processors could delay completion
  - Decreasing task times could delay completion

Learning Outcomes

- General Education Level
  - basic graph theory
  - logical thinking
  - practice with attention to detail
  - transitivity

"I expect you all to be independent, innovative, critical thinkers who will do exactly as I say!"

- Discrete Mathematics/Linear Algebra Level
  - matrix multiplication
  - posets and Hasse diagrams
  - introduction to NP-Complete problems

(More) Learning Outcomes

- Computer Programming Level
  - practice with indexing
  - learning to adhere closely to project specifications
  - using graphical display tools
  - file manipulation
  - patience

http://static.freepik.com