Mathematics for Computer Science

MIT 6.042J/18.062J

Partial Orders

 September 18, 2002

Types of Relations

Equivalence

- Reflexive, Transitive, *Symmetric*.

Partial Orders

- (Reflexive), Transitive, *Antisymmetric*.

 September 18, 2002

Antisymmetric

$$\forall a,b \quad a \neq b \land aRb \rightarrow \neg(bRa)$$

Maybe aRb or bRa, **never both.**

– Either places an order on a,b

– Or a and b are unrelated (*incomparable*)

 September 18, 2002

Partial Order (Poset (Z,≤))

$a\ R\ b$ if $a \leq b,\ a,b \in Z$

- Reflexive YES
- Transitive YES
- Symmetric NO
- Antisymmetric YES

- in fact a Total Order

 September 18, 2002

Total Order

- A relation R on a set A is a total order if it is a partial order and for any a,b in A, either aRb or bRa

  – For any two distinct integers, x and y, either $x<y$ or $y<x$

 September 18, 2002

Not a Total Order: Divisibility

$a\ R\ b$ if $a \mid b,\ \ a,b \in Z$

- 3|9, but not 9|3
- But 5 and 9 are not related (incomparable)

 September 18, 2002

Symmetric vs Antisymmetric

Symmetric:

$$\forall a,b \quad aRb \rightarrow bRa$$

Antisymmetric:

$$\forall a,b \quad a \neq b \wedge aRb \rightarrow \neg(bRa)$$

EXERCISE

- *Question:* Can R be neither symmetric nor antisymmetric?
- YES: R ::= {(1,2) (2,1) (1,4)}

- *Question:* Can R be both symmetric and antisymmetric?
- YES: R ::= {(1,1)}

A Relation on Classes

cRd ::= class *c* is *listed* as a prerequisite to class *d* in the 6-3 curriculum

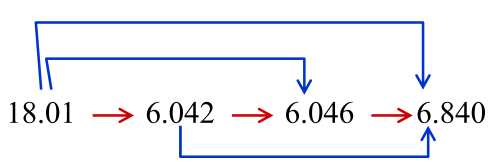
A Partial Order?

cRd ::= class *c* is *listed* as a prerequisite to class *d* in the 6-3 curriculum

- Reflexive No
- Transitive No
- Antisymmetric Yes *(certainly hope so!)*

Creating a Partial Order

- Reflexive and Transitive closure of R



A Relation on Classes

cRd ::= class *c* is *listed* as a prerequisite to class *d* in the 6-3 curriculum

Prerequisites

- 18.01 → 6.042
- 18.01 → 18.02
- 18.01 → 18.03
- 8.01 → 8.02
- 6.001 → 6.034
- 6.042 → 6.046
- 18.03, 8.02 → 6.002
- 6.001, 6.002 → 6.004
- 6.001, 6.002 → 6.003
- 6.004 → 6.033
- 6.033 → 6.857
- 6.046 → 6.840

Scheduling Problems

- How many terms will it take to graduate?
- How many classes does one need to take each term?
- Who's going to plan the whole thing?

- build a *Dependency Graph*

Identify Minimal Elements

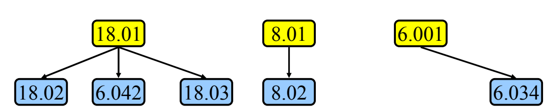
- **18.01** → 6.042
- 18.01 → 18.02
- 18.01 → 18.03
- **8.01** → 8.02
- **6.001** → 6.034
- 6.042 → 6.046
- 18.03, 8.02 → 6.002
- 6.001, 6.002 → 6.004
- 6.001, 6.002 → 6.003
- 6.004 → 6.033
- 6.033 → 6.857
- 6.046 → 6.840

Directed Acyclic Graph (DAG)

Prerequisites

- **18.01** → **6.042**
- 18.01 → **18.02**
- 18.01 → **18.03**
- **8.01** → **8.02**
- **6.001** → **6.034**
- 6.042 → 6.046
- 18.03, 8.02 → 6.002
- 6.001, 6.002 → 6.004
- 6.001, 6.002 → 6.003
- 6.004 → 6.033
- 6.033 → 6.857
- 6.046 → 6.840

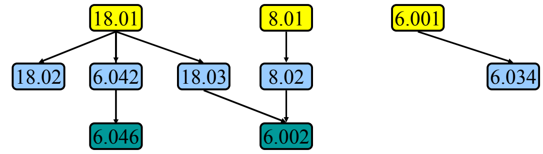
Directed Acyclic Graph (DAG)



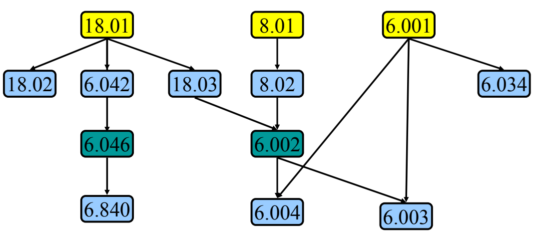
Prerequisites

- **18.01** → **6.042**
- 18.01 → **18.02**
- 18.01 → **18.03**
- **8.01** → **8.02**
- **6.001** → **6.034**
- 6.042 → **6.046**
- 18.03, 8.02 → **6.002**
- 6.001, 6.002 → 6.004
- 6.001, 6.002 → 6.003
- 6.004 → 6.033
- 6.033 → 6.857
- 6.046 → 6.840

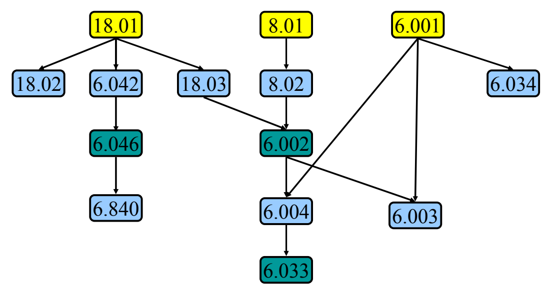
Directed Acyclic Graph (DAG)



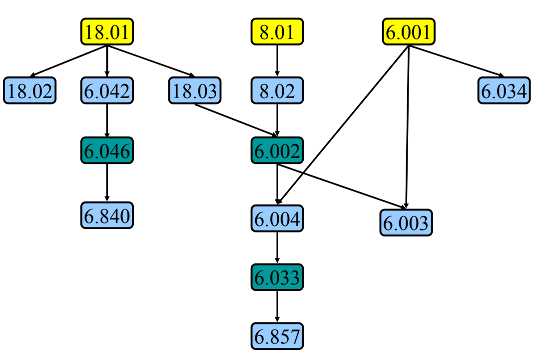
Directed Acyclic Graph (DAG)



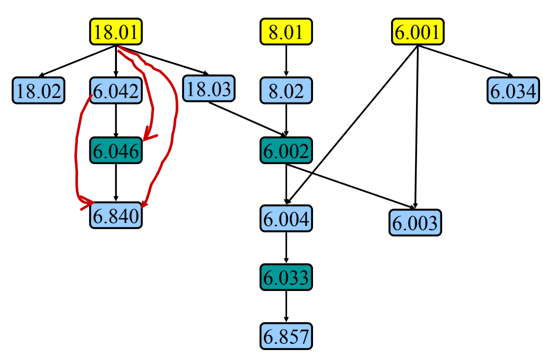
Directed Acyclic Graph (DAG)



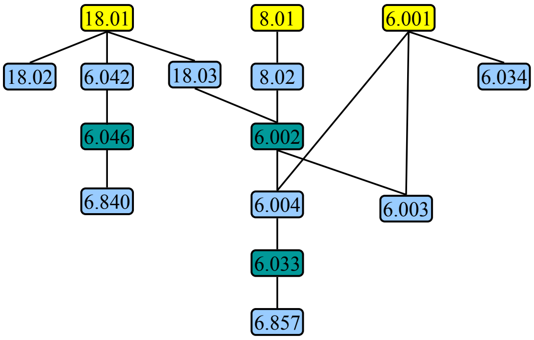
Directed Acyclic Graph (DAG)



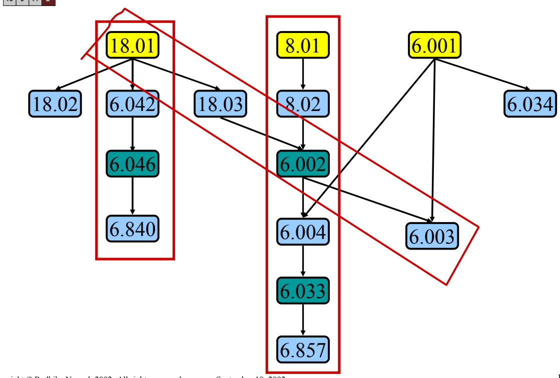
Partial Order (transitive edges)



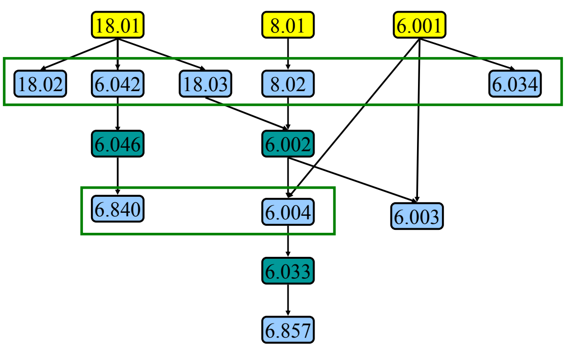
Hasse Diagram (no transitive edges)



Chains



Anti Chains



In-Class Problem 1

Scheduling Problems

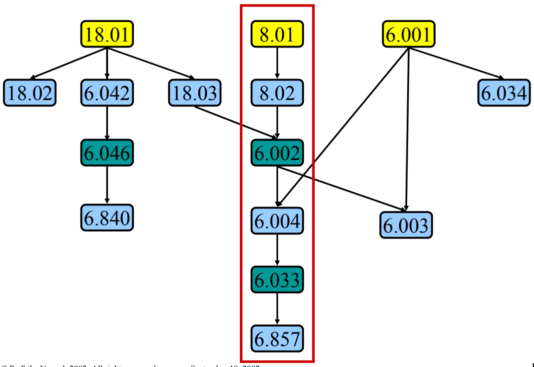
- How many terms will it take to graduate?
- How many classes does one need to take each term?
- Who's going to plan the whole thing?

Parallel Task Scheduling

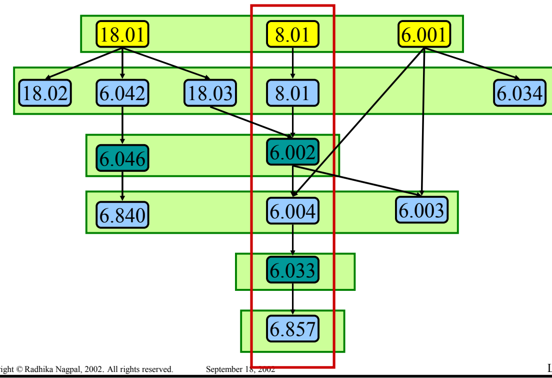
Theorem: If the longest chain has size t, then the elements can be partitioned into t antichains.

- 6 terms are **necessary** to complete the curriculum
- *and* **sufficient** (if you can take unlimited courses per year...)

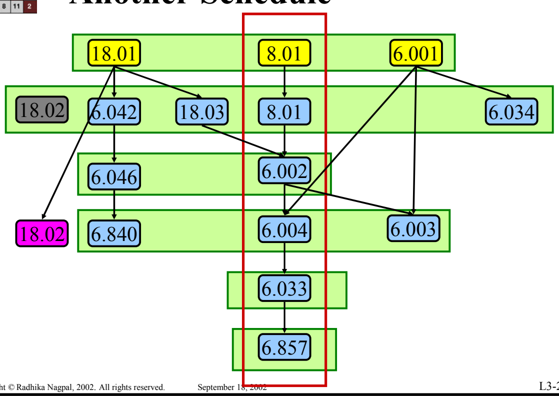
Necessary: Maximal Length Chain



Sufficient: Schedule



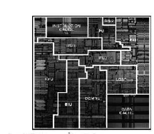
Another Schedule



Scheduling problems

- Real problem far more complex:
  - Limited per term, scheduling conflicts
- Other examples:
  - Stata Center construction
  - Program dependency graph

RS/6000 vs Alpha
(IBM vs DEC)



- Superscalar: Hardware parallelism
- Alphas: Increase pipelining and clock speed
- Longest chain = minimum time to complete the program, *no matter how parallel your machine is!*

Dilworth's Theorem

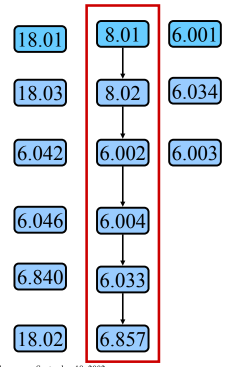
Theorem: Every poset of n elements has
- Either a chain of size at least t,
- or an antichain of size at least $\left\lceil \frac{n}{t} \right\rceil$

for all $1 \le t \le n$.

Prereqs Graph

- Total **15** classes
- Length of maximum chain = **6**
- No chain of length 7
- → must be ***at least one antichain*** with size at least **3**.
  - *(at least one term where you have to take ≥3 classes)*

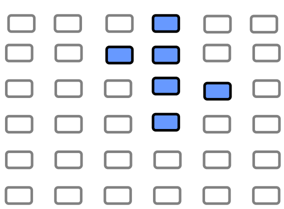
Classes per Term



Dilworth's Theorem: Example

Consider a set S, where $|S| = n = 6$

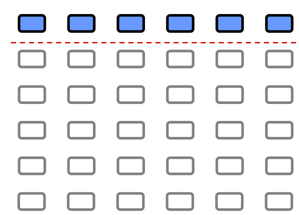
Column= chain, Row = anti-chain



Dilworth's Theorem: Example

$|S| = n = 6$

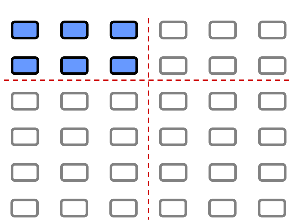
Longest chain $t = 1$



Dilworth's Theorem: Example

$|S| = n = 6$

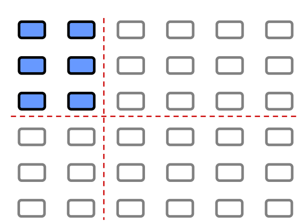
Longest chain $t = 2$



Dilworth's Theorem: Example

$|S| = n = 6$

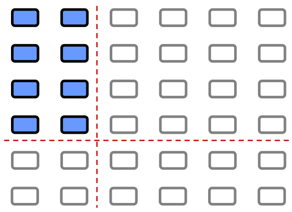
Longest chain $t = 3$



Dilworth's Theorem: Example

$|S| = n = 6$

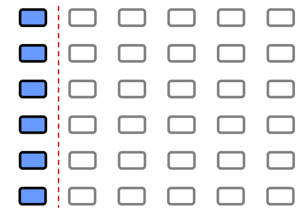
Longest chain $t = 4$



Dilworth's Theorem: Example

$|S| = n = 6$

Longest chain $t = 6$



Dilworth's Theorem

Theorem: Every poset of n elements has

- Either a chain of size at least t,
- or an antichain of size at least $\left\lceil \frac{n}{t} \right\rceil$

for all $1 \le t \le n$.

In-Class Problems