Example Class 4

Set Theory & Relations

Outline

- Mathematicians and Chess Players
- Lexicographic Order
- Topological Sort

http://www.publicdomainpictures.net/pictures/80000/nahled/chess-pieces-clipart.jpg

Mathematicians and Chess Players (I)

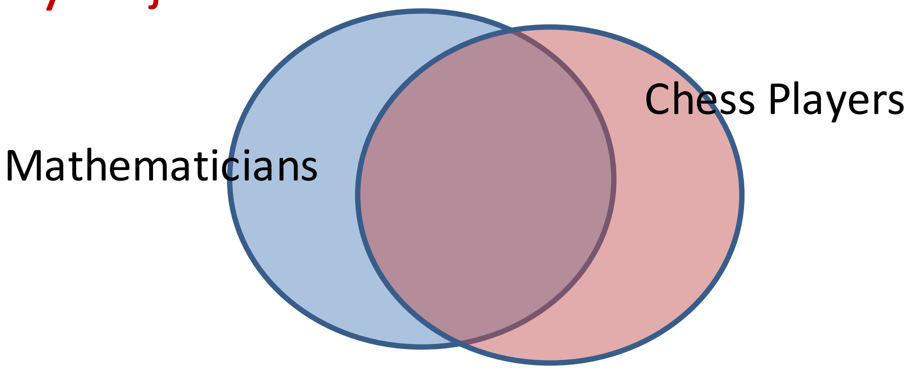
Can the oldest mathematician among chess players and the oldest chess player among mathematicians be 2 different people?

Reference: "Basic Set Theory" by A. Shen

Mathematicians and Chess Players (I)

Can the oldest mathematician among chess players and the oldest chess player among mathematicians be 2 different people?

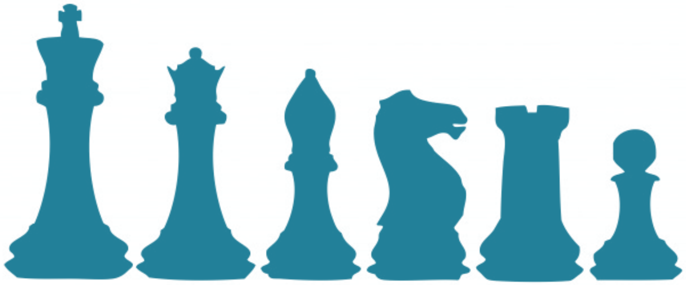
No: look at the age of people in {Mathematicians} ∩ {Chess Players}



Reference: "Basic Set Theory" by A. Shen

Mathematicians and Chess Players (II)

Can the best mathematician among chess players and the best chess player among mathematicians be 2 different people?

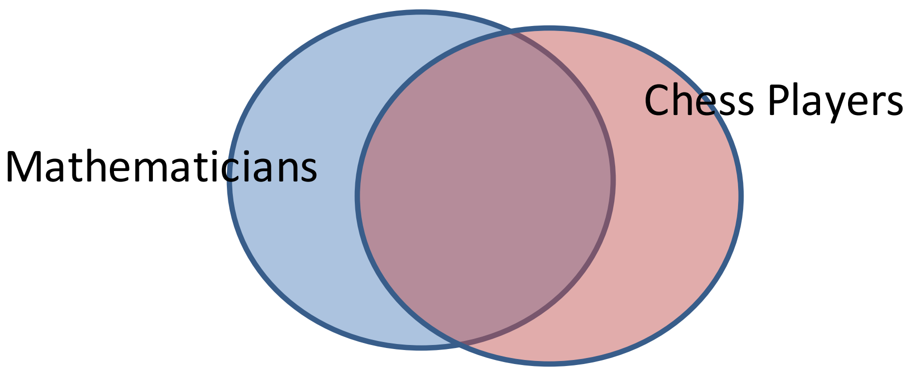


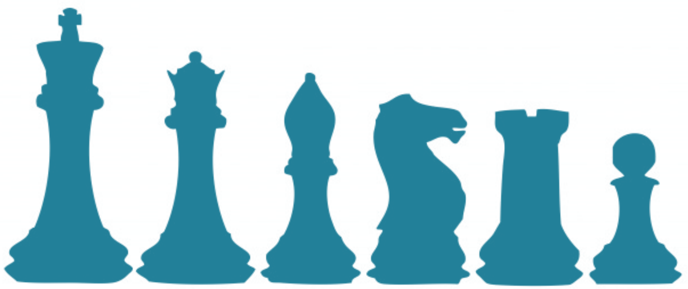
Reference: "Basic Set Theory" by A. Shen

Mathematicians and Chess Players (II)

Can the best mathematician among chess players and the best chess player among mathematicians be 2 different people?

Yes: we still look at people in {Mathematicians} ∩ {Chess Players}, but we look at skills for maths and skills for chess!





Reference: "Basic Set Theory" by A. Shen

Posets

A set S together with a partial order R is called a partially ordered set, or poset (S,R).

We write a ≼ b to say that (a,b) is a partial order and a ≺ b when a is different from b.

If (S, ≼) is a poset, and every two elements of S satisfy that either a ≼ b or b ≼ a, then S is called a totally ordered set, and ≼ is called a total order.

Example of totally ordered set? (Z, ≤)

Lexicographic Order (I)

Given two posets $(A, \leqslant_1)$ and $(B, \leqslant_2)$, the lexicographical order $\leqslant$ on the Cartesian product $A \times B$ is defined as

$$(a,b) \leqslant (a',b') \text{ if and only if } a <_1 a' \text{ or } (a = a' \text{ and } b \leqslant_2 b').$$

- What is the connection between this definition of lexicographical order and the usual one?

- Generalize this definition to the Cartesian product of n sets.

Lexicographic Order (I)

Given two posets $(A, \leqslant_1)$ and $(B, \leqslant_2)$, the lexicographical order $\leqslant$ on the Cartesian product $A \times B$ is defined as

$$(a,b) \leqslant (a',b') \text{ if and only if } a \prec_1 a' \text{ or } (a = a' \text{ and } b \leqslant_2 b').$$

- What is the connection between this definition of lexicographical order and the usual one?

The usual one is alphabetical order, starting with the 1st character in the string, then the 2nd (if the first was equal) etc., considering "no character" for shorter words to be less than "a". It is a particular case of this definition.

- Generalize this definition to the Cartesian product of n sets.

Define $\leqslant$ on $A_1 \times \ldots \times A_n$ by $(a_1,\ldots a_n) \prec (b_1,\ldots,b_n)$ if $a_1 \prec b_1$ or $a_1 = b_1,\ldots, a_i = b_i$ and $a_{i+1} \prec b_{i+1}$

Lexicographic Order (II)

Given two posets $(A, \preccurlyeq_1)$ and $(B, \preccurlyeq_2)$, the lexicographical order $\preccurlyeq$ on the Cartesian product $A \times B$ is defined as

$$(a,b) \preccurlyeq (a',b') \text{ if and only if } a \prec_1 a' \text{ or } (a = a' \text{ and } b \preccurlyeq_2 b').$$

- Is the lexicographic order a partial order?
- Is the lexicographic order a total order?

Lexicographic Order (II)

Given two posets $(A, \preccurlyeq_1)$ and $(B, \preccurlyeq_2)$, the lexicographical order $\preccurlyeq$ on the Cartesian product $A \times B$ is defined as

$$(a,b) \preccurlyeq (a',b') \text{ if and only if } a \prec_1 a' \text{ or } (a = a' \text{ and } b \preccurlyeq_2 b').$$

- Is the lexicographic order a partial order? Yes it is!
- Is the lexicographic order a total order? If A and B are totally ordered, then yes, otherwise not necessarily.

Lexicographic Order (III)

Given two posets (A, $\preccurlyeq_1$) and (B, $\preccurlyeq_2$), the lexicographical order $\preccurlyeq$ on the Cartesian product $A \times B$ is defined as

$(a,b) \preccurlyeq (a',b')$ if and only if $a \prec_1 a'$ or ($a = a'$ and $b \preccurlyeq_2 b'$).

- Let A=(a,b,c) with a $\preccurlyeq_1$ b $\preccurlyeq_1$ c and let B = {0,1} with 0 $\preccurlyeq_2$ 1. Compute the lexicographic ordering on AxB.

Lexicographic Order (III)

Given two posets $(A, \preccurlyeq_1)$ and $(B, \preccurlyeq_2)$, the lexicographical order $\preccurlyeq$ on the Cartesian product $A \times B$ is defined as

$$(a,b) \preccurlyeq (a',b') \text{ if and only if } a \prec_1 a' \text{ or } (a = a' \text{ and } b \preccurlyeq_2 b').$$

- Let A=(a,b,c) with $a \preccurlyeq_1 b \preccurlyeq_1 c$ and let B = {0,1} with $0 \preccurlyeq_2 1$. Compute the lexicographic ordering on AxB.

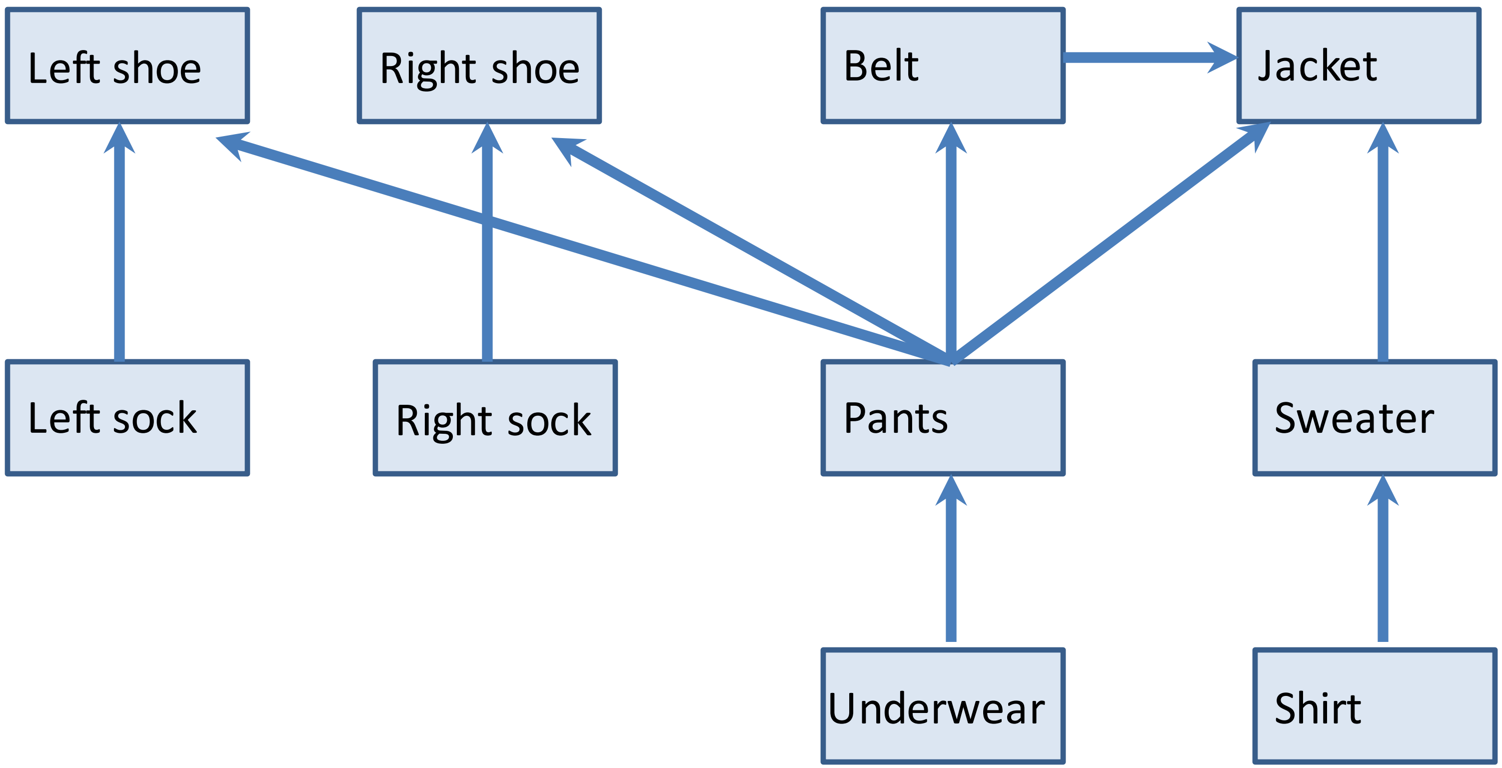
The lexicographic ordering on AxB is:
$(a,0) \preccurlyeq (a,1) \preccurlyeq (b,0) \preccurlyeq (b,1) \preccurlyeq (c,0) \preccurlyeq (c,1)$

Scheduling

- Suppose you have a set A of tasks, and a set of constraints specifying that a given task depends on the completion of another task.
- You are interested in a scheduling, that is an order in which to perform all the tasks one at a time while respecting the dependency constraints.
- Finding a total ordering that is consistent with a partial order is called topological sorting.

A topological sort of a partial order $\leqslant$ on a set A is a total order [on A such that $a \leqslant b$ implies a [b.

Topological Sort (I)



Compute a topological sort.

http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-042j-mathematics-for-computer-science-spring-2010/readings/MIT6_042JS10_chap07.pdf

Topological Sort (II)

An example (not unique) of topological sort is:

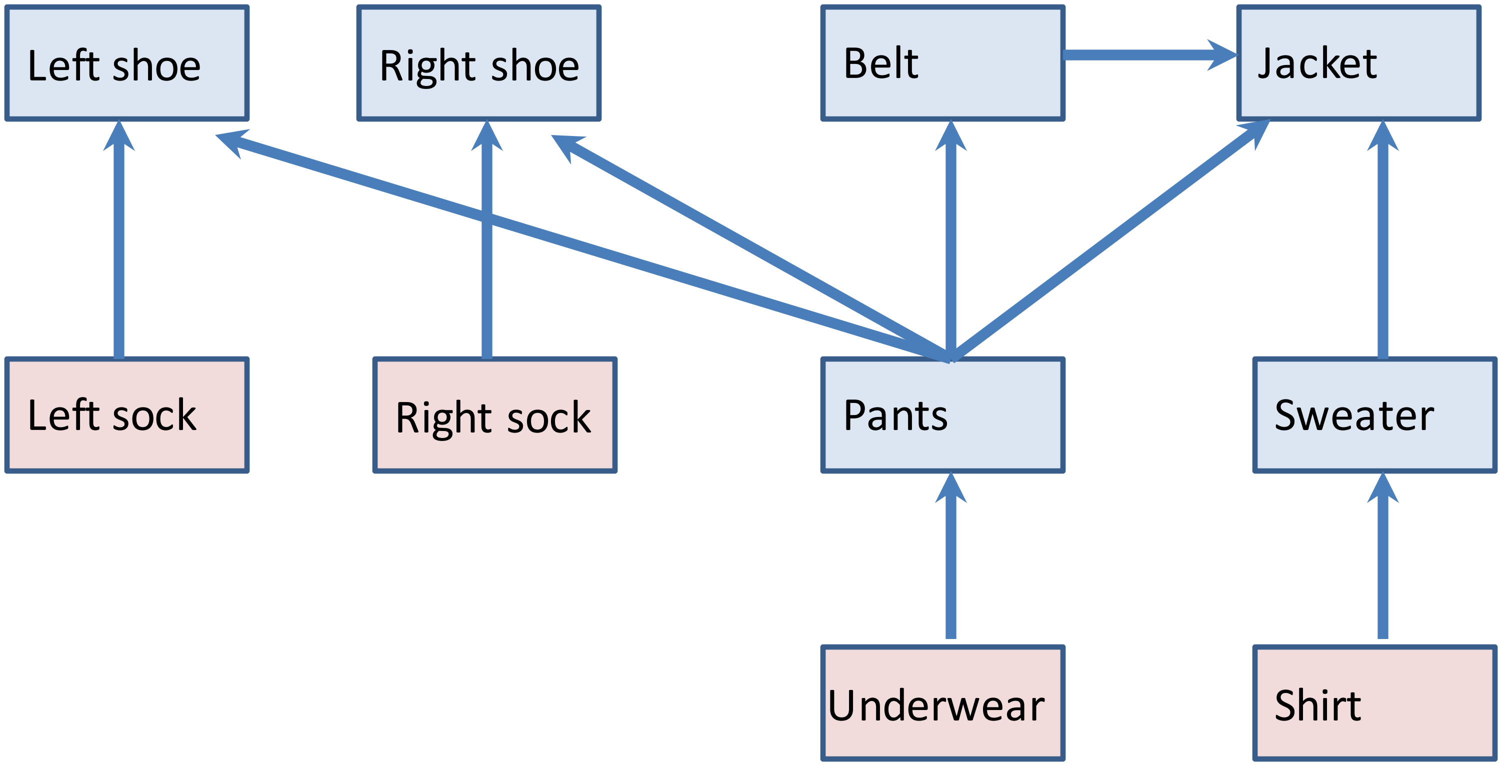
shirt [sweater [underwear [left sock [right sock [pants [left shoe [right shoe [belt [jacket

Let $\preccurlyeq$ be a partial order on the set A. An element a in A is minim**um** iff it is $\preccurlyeq$ every other element of A.

The element a is minim**al** iff no other element is $\preccurlyeq$ a.

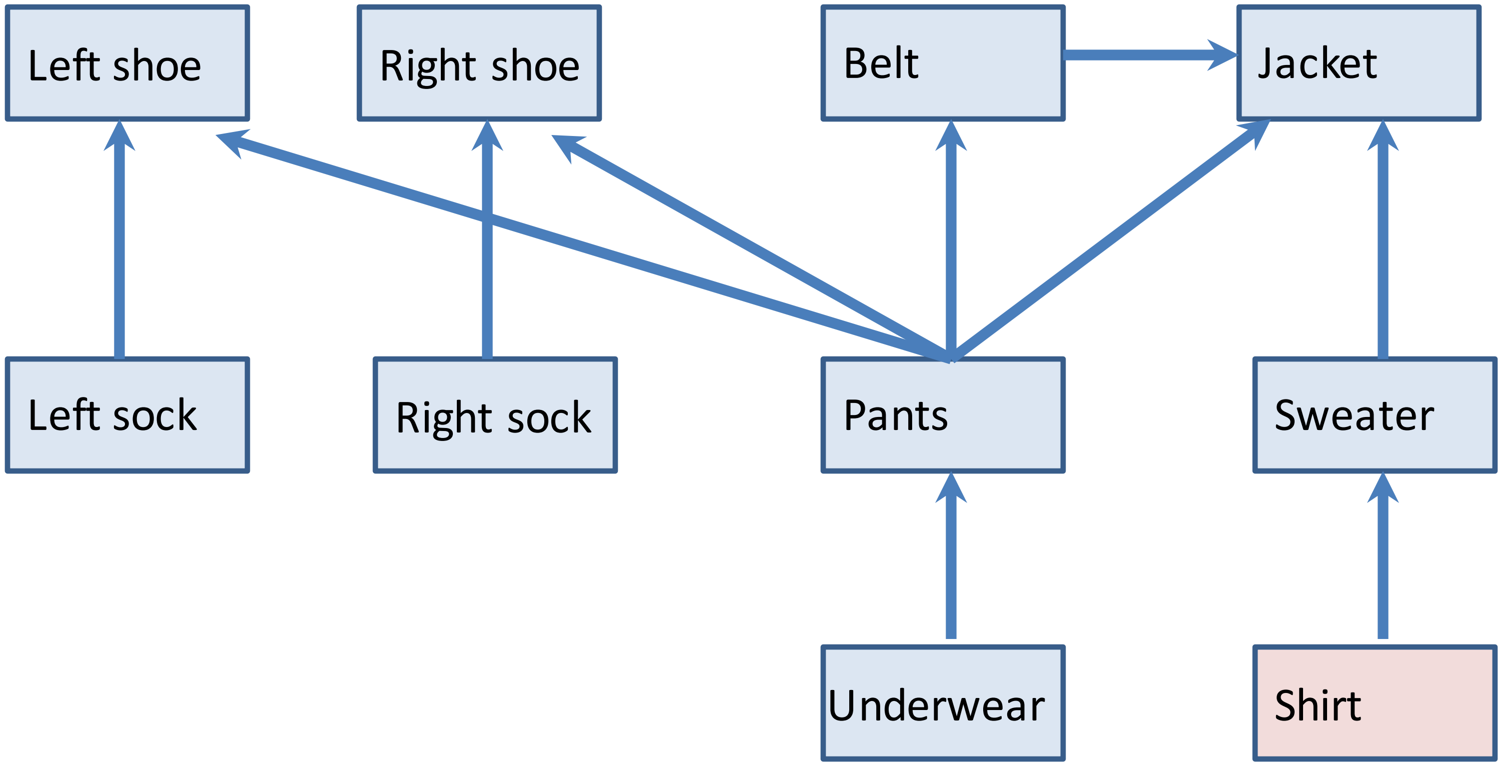
Minimum = minimal in total order but not otherwise!

Topological Sort (III)



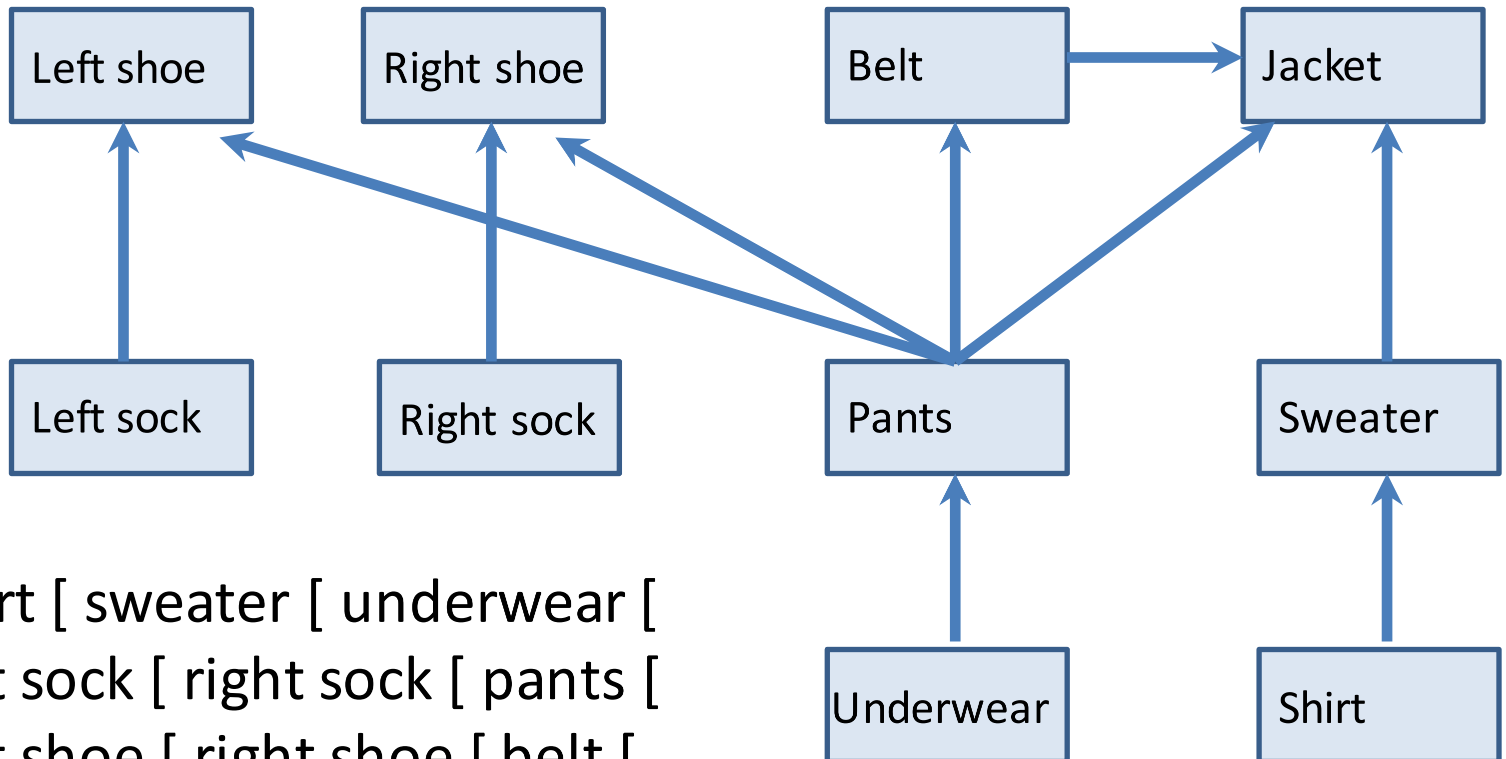
There are 4 minimal elements among the clothes: left sock, right sock, underwear, shirt. Pick one of these minimal elements, say shirt.

Topological Sort (IV)



Then pick a minimal element among the remaining ones (once shirt is removed, sweater becomes minimal) and iterate.

Topological Sort (V)



shirt [sweater [underwear [left sock [right sock [pants [left shoe [right shoe [belt [jacket

http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-042j-mathematics-for-computer-science-spring-2010/readings/MIT6_042JS10_chap07.pdf