Given a finite poset $P = (X, \preceq)$, a natural question to ask is “can we find a linear order $L$ that includes $P$?” It turns out that the answer is Yes, we always can. In fact we can go further.

(Why is this a natural question? Because one very common application of partial orderings is to represent the relationships between subtasks in a large project ... either something like the Hallowe’en example, or something like a dependency diagram for a large software system. Frequently we need to process every item in the set, but we only have one processor (in the Hallowe’en example the processor was me, trying to get all the subtasks completed). We need to put the full set of items into a sequence so that the lone processor can work through them without ever trying to do a subtask before its predecessors have been completed.)

First, a definition: Let $P = (X, \preceq)$ be a poset. A **linear extension** of $P$ is a linear order $L = (X, \preceq')$ where $\preceq \subseteq \preceq'$

The last three symbols on that line may look they can’t possibly be meaningfully used in that way. But remember that a partial order is just a relation, and a relation is just a set of ordered pairs. So when we write $\preceq \subseteq \preceq'$, we mean “the set of ordered pairs that defines $\preceq$” is a subset of “the set of ordered pairs that defines $\preceq'$”

This definition of **linear extension** is our way of formalizing the concept of a linear order including a particular poset. If $L$ is a linear extension of $P$, then $L$ includes all the relationships in $P$, and also enough new relationships to eliminate all incomparabilities.
**Theorem:** Let $P = (X, \preceq)$ be a finite partial order in which $x$ and $y$ are incomparable. Then there exists a linear extension $L_1$ of $P$ such that in $L_1$, $x \prec y$. There also exists a linear extension $L_2$ of $P$ such that in $L_2$, $y \prec x$.

**Proof:** Let $P = (X, \preceq)$ be a finite partial order in which $x$ and $y$ are incomparable (that is, neither $x \preceq y$ nor $y \preceq x$).

We define a new partial order $\preceq'$ on $X$ as follows:

- $x \preceq' y$
- if $a \preceq b$ then $a \preceq' b$
- if $a \preceq x$ and $y \preceq b$ then $a \preceq' b$

Ok, $\preceq'$ certainly includes $x \preceq' y$, and it includes all of $\preceq$ ... but is it a partial order?

Reflexive: $\preceq'$ is reflexive because $\preceq$ is reflexive.

Anti-symmetric: Suppose $a \preceq' b$ and $b \preceq' a$. We need to show $a = b$.

- $a \preceq' b$ can be true in two ways: either $a \preceq b$ or $a \preceq x$ and $y \preceq b$.
- $b \preceq' a$ can be true in two ways: either $b \preceq a$ or $b \preceq x$ and $y \preceq a$.

This gives us four cases to consider in order to show that $a = b$. 
<table>
<thead>
<tr>
<th>Case</th>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a \leq b$ and $b \leq a$</td>
<td>In this case $a = b$ because $\leq$ is anti-symmetric</td>
</tr>
<tr>
<td>2</td>
<td>$a \leq b$ and $b \leq x$ and $y \leq a$</td>
<td>We can combine these three facts to get $y \leq a \leq b \leq x$ from which we see $y \leq x$ ... but this is impossible since $x$ and $y$ are incomparable. Thus this case cannot exist</td>
</tr>
<tr>
<td>3</td>
<td>$b \leq a$ and $a \leq x$ and $y \leq b$</td>
<td>As for Case 2, this case cannot exist</td>
</tr>
<tr>
<td>4</td>
<td>$a \leq x$ and $y \leq b$ and $b \leq x$ and $y \leq a$</td>
<td>From $y \leq a$ and $a \leq x$ we get $y \leq x$, which is impossible ... so this case cannot exist either.</td>
</tr>
</tbody>
</table>

Thus only Case 1 is possible, and in Case 1 we saw $a = b$

Therefore $a = b$, and therefore $\preceq'$ is anti-symmetric.

Proof that $\preceq'$ is transitive follows a similar pattern. (Exercise: complete this part of the proof.)

Thus $\preceq'$ is a partial order on $X$, and it contains fewer incomparable pairs than $\leq$ does. If $\preceq'$ is a linear order (no incomparable pairs at all) then we have extended $\preceq$ to a linear order, as required. If $\preceq'$ is not a linear order, we can choose an incomparable pair $w$ and $z$ and extend $\preceq'$ to $\preceq''$ ... and if $\preceq''$ is not a linear order, we can extend it to $\preceq'''$, etc. Eventually we must eliminate all incomparable pairs, ending up with a linear order that contains the ordered pair $(x,y)$ as stated in the theorem.

Now by switching $x$ and $y$ throughout all of the preceding argument, we will get a linear order that contains $(y,x)$ as an ordered pair.

Thus the theorem is proved.
As we now know, every poset that is not a linear order can be extended to at least two different linear extensions.

Observation: Let $P = (X, \leq)$ be a poset. Let $x$ and $y$ be distinct elements of $X$. If $x \prec y$, then $x$ precedes $y$ in all linear extensions of $P$. Conversely, if $x$ and $y$ are incomparable elements of $X$, there exists at least one linear extension of $P$ in which $x \prec y$, and at least one linear extension of $P$ in which $y \prec x$. Our proof of the previous theorem justifies this observation.

This suggests an interesting question: if we are given the set of all linear extensions of a poset, can we reverse-engineer it and determine the original poset?

The answer is yes!

For each pair $x$ and $y$, we can check to see if $x$ precedes $y$ in all the linear extensions ... if so, we know $x \prec y$. Similarly if $y$ precedes $x$ in all the linear extensions, we know $y \prec x$. However, if there is at least one linear extension in which $x$ precedes $y$, and at least one in which $y$ precedes $x$, then $x$ and $y$ are incomparable. Thus we can determine exactly which pairs are related, and how they are related.
The next question is: do we always need the full set of linear extensions of a poset to
reconstruct the poset?

The answer (which surprised me the first time I learned it) is no!

Consider this example:

The poset on the left has exactly three linear extensions – they are shown on the right. But we
can reconstruct the poset using just the first and third linear extensions:

To see that these give all the
information needed, let’s look at b and c. We see that b precedes c in both linear
extensions, so we know $b \prec c$. We can deduce the same thing
for pairs such as d and e, a and b, etc.

But consider b and d. B
precedes d in one of the linear
extensions, and d precedes b in
the other. Thus we know b and
d are incomparable in P. Similarly we can discover that c and d are incomparable in P.
Definition: If \( P \) is a poset and \( R = \{L_1, L_2, \ldots, L_i\} \) is a set of linear extensions of \( P \) that allows us to completely reconstruct \( P \), we call \( R \) a realizer for \( P \).

Note that every poset has at least one realizer since the full set of linear extensions of any poset is a realizer for that poset.

Definition: We have seen that for a given poset, some realizers are smaller than others. Let \( P \) be a poset and let \( R \) be a realizer for \( P \) with the property that there is no smaller realizer for \( P \). If the number of extensions in \( R \) is \( k \), then we say that \( P \) has \textbf{dimension} \( k \) (which we sometimes write as \( \dim P = k \))

For the poset in the example above, we can see there is no realizer of size 1 (because we need at least two extensions to show that \( b \) and \( d \) are incomparable). Since we have found a realizer of size 2, we now know that the dimension of this poset is 2.
So, how high can the dimension of a poset be? Consider this example:

The poset is shown on the left ... none of the elements are related to each other. Since any linear order of \{a, b, c, d, e\} is a valid linear extension of this poset, we see that there are 5! = 120 linear extensions of this poset. And yet the two extensions on the right in the diagram are a realizer for the poset! (Make sure you see why). So once again, this poset has dimension 2.

Another example:

Again we see that the two linear extensions on the right form a realizer for the poset, (and again we see that the dimension of the poset must be > 1 since there are incomparable elements) so we see this poset also has dimension 2.
Perhaps you are forming an hypothesis that all posets have dimension = 1 (which is the case if the poset is in fact a linear order) or dimension = 2 ... which is the case for all the other posets we have looked at.

Sadly the situation is more complicated than that. Consider *this* example:

I claim that the poset shown here has dimension at least 3. We need at least one linear extension that has b1 preceding a1 ... but in this linear extension, we must have a2 preceding b2, and a3 preceding b3. Therefore we need another linear extension with b2 preceding a2 ... and in this one, a3 must precede b3. Therefore we need another linear extension with b3 preceding a3.

Three such linear extensions are shown in the diagram, and we can see that these form a realizer for this poset. Therefore this poset has dimension 3.

We can generalize this example to have elements $a_1$ to $a_n$ and $b_1$ to $b_n$ - these are called the **standard examples** - the one shown above is called $S_3$ because it has 3 elements in each of its two subsets.

It turns out that the standard example of size $n$ (ie. $S_n$) has dimension $n$.

Thus there is no limit to how large the dimension of a poset can be.
Let’s consider one more question – in some ways, the most interesting of all: Given a poset \( P \), can we determine its dimension?

If \( P \) is a linear order, \( \dim P = 1 \)

We can determine if \( \dim P = 2 \) using an algorithm that is (unfortunately) outside the scope of this course. This algorithm runs fairly quickly.

But beyond that ... nobody knows if there exists any efficient way to answer the question “Does \( \dim P = k \)?” where \( k \) is any integer \( \geq 3 \). In fact, this has been identified as one of the most difficult questions to answer in all of computational mathematics.

This is a tiny foretaste of a topic that will be explored in future courses. The world of problems that we can solve using computers seems to split into two classes: problems that we can solve quickly, and problems that are so difficult that we don’t believe we will ever be able to solve them quickly. And sometimes the difference between the two classes is as simple as changing a 2 to a 3. People have spent decades trying to figure out why.