CS 207: Discrete Structures

Instructor : S. Akshay

Aug 6, 2015

Lecture 09 – Basic mathematical structures: posets, chains, antichains and their applications

Recall: Partial Orders and Equivalence relations

- Reflexive: $\forall a \in S$, aRa.
- Symmetric: $\forall a, b \in S$, aRb implies bRa.
- Anti-symmetric: $\forall a, b \in S$, aRb, bRa implies $a = b$.
- Transitive: $\forall a, b, c \in S$, aRb, bRc implies aRc.

	Reflexive	Transitive	Symmetric	Anti-symmetric
Equivalence	✓	✓	✓	
Partial order	✓	✓		✓

A total order is a partial order $\preceq$ on S in which every pair of elements is comparable, i.e., $\forall a, b \in S$, either $a \preceq b$ or $b \preceq a$.

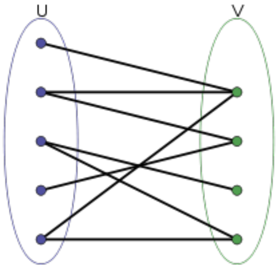
Definition (poset)

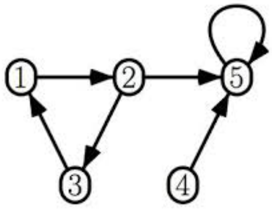
A set S together with a partial order $\preceq$ on S, is called a partially-ordered set or poset, denoted $(S, \preceq)$.
Examples: $(\mathbb{Z}, \leq)$, $(\mathcal{P}(S), \subseteq)$, $(\mathbb{Z}^+, \mid\ )$.

Graphical representation of relations: posets

Recall: any relation on a set can be represented as a graph with

- nodes as elements of the set and
- directed edges between them indicating the ordered pairs that are related.





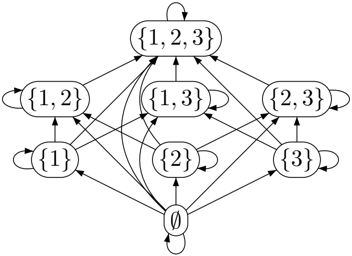
- Did these come from posets?
- Do graphs defined by posets have any "special" properties?

Graphical representation of relations: posets

- Let $S = \{1, 2, 3\}$. Recall the poset $(\mathcal{P}(S), \subseteq)$.
- How does the graph of $(\mathcal{P}(S), \subseteq)$ look like?

Graphical representation of relations: posets

- Let $S = \{1, 2, 3\}$. Recall the poset $(\mathcal{P}(S), \subseteq)$.
- How does the graph of $(\mathcal{P}(S), \subseteq)$ look like?



Graphical representation of relations: posets

- Let $S = \{1, 2, 3\}$. Recall the poset $(\mathcal{P}(S), \subseteq)$.
- How does the graph of $(\mathcal{P}(S), \subseteq)$ look like?

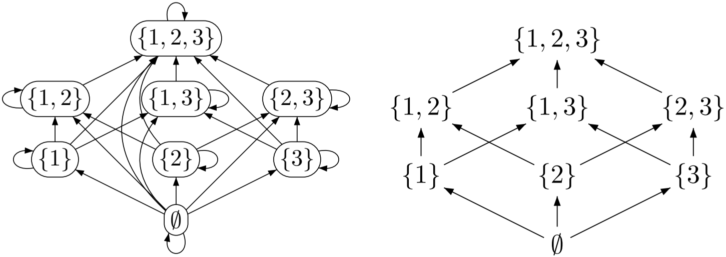


Figure : Graph of a poset and its Hasse diagram

Graphical representation of relations: posets

- Let $S = \{1, 2, 3\}$. Recall the poset $(\mathcal{P}(S), \subseteq)$.

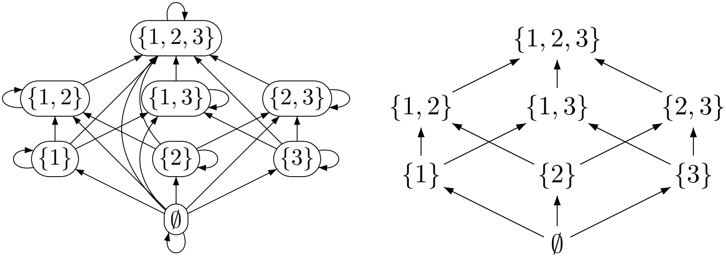


Figure : Graph of a poset and its Hasse diagram

- What is "special" about these graphs?

Graphical representation of relations: posets

- Let $S = \{1, 2, 3\}$. Recall the poset $(\mathcal{P}(S), \subseteq)$.

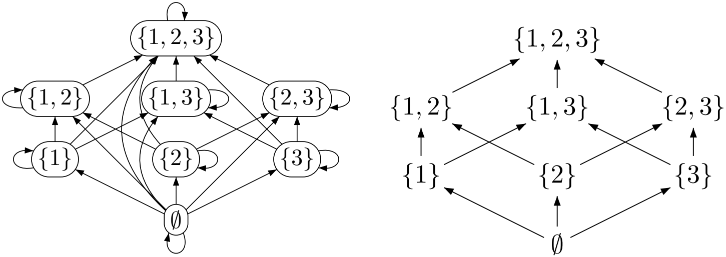


Figure : Graph of a poset and its Hasse diagram

- What is "special" about these graphs?
- Graphs of posets are "acyclic" (except for self-loops).
- Starting from a node and following the directed edges (except self-loops), one can't come back to the same node.

Graphical representation of relations: posets

- Let $S = \{1,2,3\}$. Recall the poset $(\mathcal{P}(S), \subseteq)$.

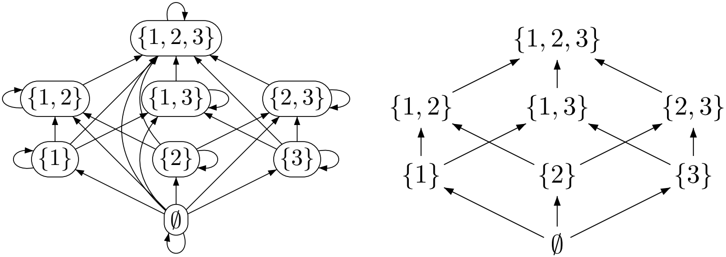


Figure : Graph of a poset and its Hasse diagram

- What is "special" about these graphs?
- Graphs of posets are "acyclic" (except for self-loops).
- Starting from a node and following the directed edges (except self-loops), one can't come back to the same node.
- Given the Hasse diagram of a poset, its reflexive transitive closure gives back the graph of the poset.

Chains and Anti-chains

Definition

Let $(S, \preceq)$ be a poset. A subset $B \subseteq S$ is called

- a chain if every pair of elements in B is related by $\preceq$.
- That is, $\forall a, b \in B$, we have $a \preceq b$ or $b \preceq a$ (or both).

Chains and Anti-chains

Definition

Let $(S, \preceq)$ be a poset. A subset $B \subseteq S$ is called

- a chain if every pair of elements in B is related by $\preceq$.
- That is, $\forall a, b \in B$, we have $a \preceq b$ or $b \preceq a$ (or both).
- Thus, $\preceq$ is a total order on B.

Chains and Anti-chains

Definition

Let $(S, \preceq)$ be a poset. A subset $B \subseteq S$ is called

- a chain if every pair of elements in B is related by $\preceq$.
- That is, $\forall a, b \in B$, we have $a \preceq b$ or $b \preceq a$ (or both).
- Thus, $\preceq$ is a total order on B.

Definition

Let $(S, \preceq)$ be a poset. A subset $A \subseteq S$ is called

- an anti-chain if no two distinct elements of A are related to each other under $\preceq$.

Chains and Anti-chains

Definition

Let $(S, \preceq)$ be a poset. A subset $B \subseteq S$ is called

- a chain if every pair of elements in B is related by $\preceq$.
- That is, $\forall a, b \in B$, we have $a \preceq b$ or $b \preceq a$ (or both).
- Thus, $\preceq$ is a total order on B.

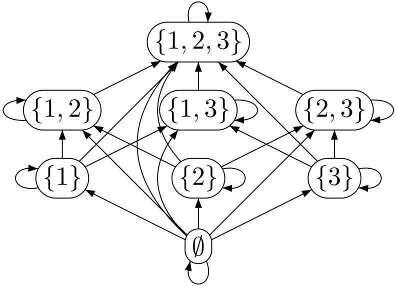
Definition

Let $(S, \preceq)$ be a poset. A subset $A \subseteq S$ is called

- an anti-chain if no two distinct elements of A are related to each other under $\preceq$.
- That is, $\forall a, b \in A, a \neq b$, we have neither $a \preceq b$ nor $b \preceq a$.

Chains and Anti-chains: examples

- Let $S = \{1, 2, 3\}$.



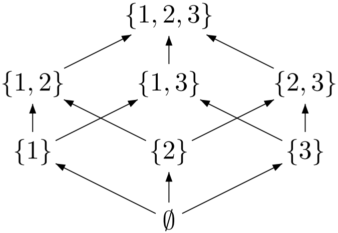


Figure : Graph of poset $(\mathcal{P}(S), \subseteq)$ and its Hasse diagram

- What are the chains in this poset?

Chains and Anti-chains: examples

- Let $S = \{1, 2, 3\}$.

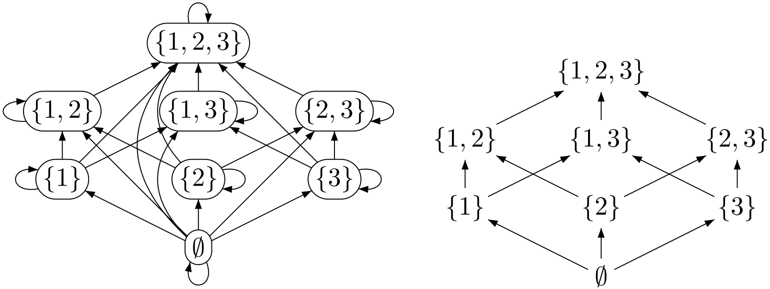


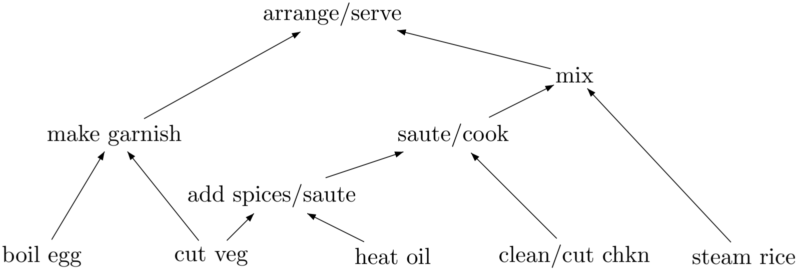
Figure : Graph of poset $(\mathcal{P}(S), \subseteq)$ and its Hasse diagram

- What are the chains in this poset?
- What are the anti-chains in this poset?

Examples and applications

A task scheduling example

Let us represent a recipe for making Chicken Biriyani as a poset!

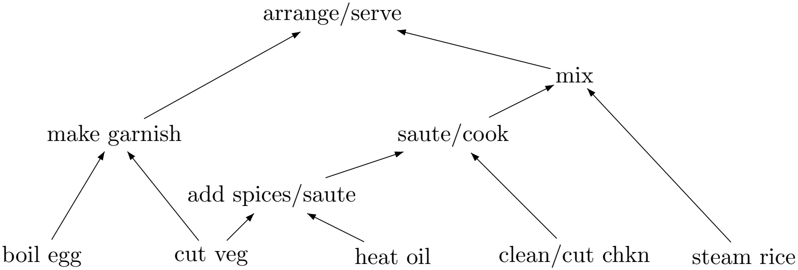


- Clearly, this shows the dependencies.

Examples and applications

A task scheduling example

Let us represent a recipe for making Chicken Biriyani as a poset!

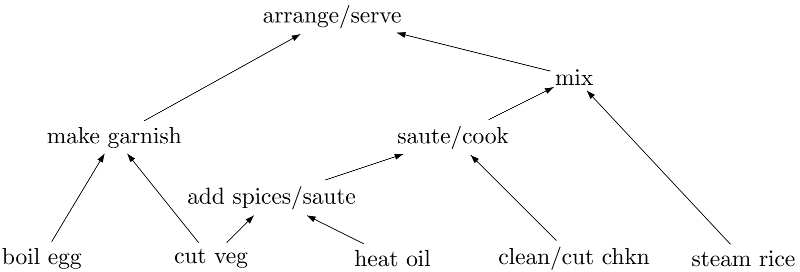


- Clearly, this shows the dependencies.
- But when you cook you need a total order, right?

Examples and applications

A task scheduling example

Let us represent a recipe for making Chicken Biriyani as a poset!



- Clearly, this shows the dependencies.
- But when you cook you need a total order, right?
- Further, this total order must be consistent with the po.
- This is called a linearization or a topological sorting.

Topological sorting

Definition

A topological sort or a linearization of a poset $(S, \preceq)$ is a poset $(S, \preceq_t)$ with a total order $\preceq_t$ such that $x \preceq y$ implies $x \preceq_t y$.

Topological sorting

Definition

A topological sort or a linearization of a poset $(S, \preceq)$ is a poset $(S, \preceq_t)$ with a total order $\preceq_t$ such that $x \preceq y$ implies $x \preceq_t y$.

Theorem

Every finite poset has a topological sort.

Topological sorting

Definition

A topological sort or a linearization of a poset $(S, \preceq)$ is a poset $(S, \preceq_t)$ with a total order $\preceq_t$ such that $x \preceq y$ implies $x \preceq_t y$.

Theorem

Every finite poset has a topological sort.

Proof: (H.W)

- First prove the following lemma:
  - Every finite non-empty poset has at least one minimal element (x is minimal if $\nexists y, y \preceq x$).

Topological sorting

Definition

A topological sort or a linearization of a poset $(S, \preceq)$ is a poset $(S, \preceq_t)$ with a total order $\preceq_t$ such that $x \preceq y$ implies $x \preceq_t y$.

Theorem

Every finite poset has a topological sort.

Proof: (H.W)

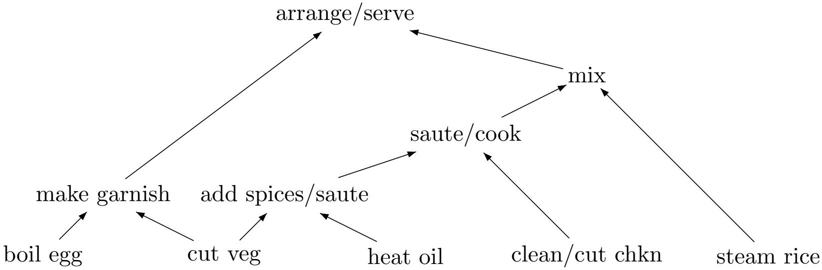
- First prove the following lemma:
  - Every finite non-empty poset has at least one minimal element (x is minimal if $\nexists y, y \preceq x$).

  What about infinite posets?
- Then, construct the chain to complete the proof.

Parallel Task Scheduling and chains

Coming back to our example,

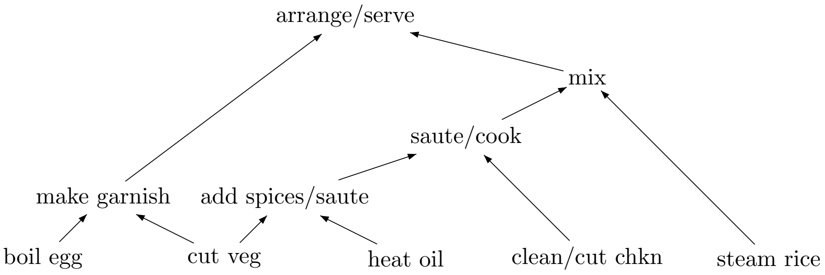
- What if there are many cooks, i.e., parallel processors?
- How do we schedule the tasks to minimize time used?



Parallel Task Scheduling and chains

Coming back to our example,

- What if there are many cooks, i.e., parallel processors?
- How do we schedule the tasks to minimize time used?

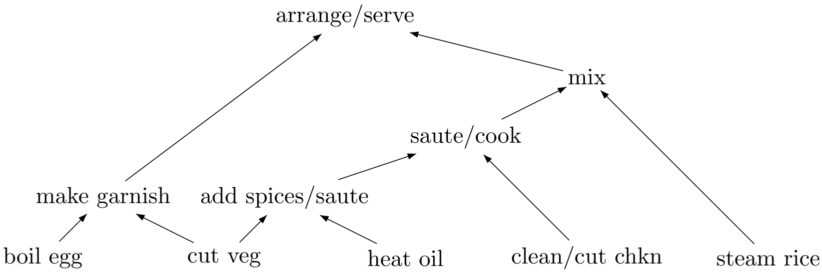


- Assume that every task takes 1 time unit.

Parallel Task Scheduling and chains

Coming back to our example,

- What if there are many cooks, i.e., parallel processors?
- How do we schedule the tasks to minimize time used?

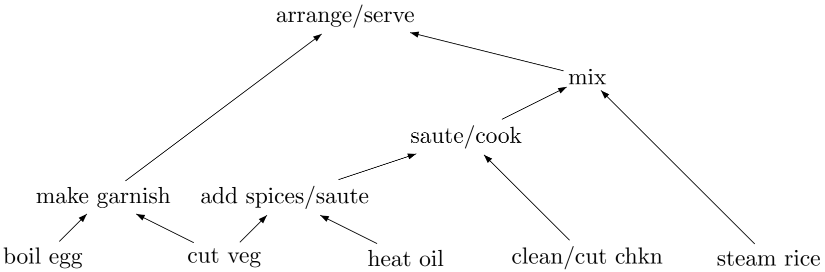


- Assume that every task takes 1 time unit.
- Clearly, we still need at least 5 time units.

Parallel Task Scheduling and chains

Coming back to our example,

- What if there are many cooks, i.e., parallel processors?
- How do we schedule the tasks to minimize time used?



- Assume that every task takes 1 time unit.
- Clearly, we still need at least 5 time units.
- That is, the length of the longest chain (length of chain = no. of elements in it).

Parallel task scheduling

For any poset, there is a legal parallel schedule that runs in t steps, where t is the length of the longest chain.

We will in fact prove:

Theorem

For a finite poset $(S, \preceq)$ with length of longest chain $= t$, we can partition S into t subsets $S_1, \ldots, S_t$ such that $\forall i \in \{1, \ldots, t\}$, $\forall a \in S_i$, if $b \preceq a, b \neq a$ then $b \in S_1 \cup \ldots \cup S_{i-1}$.

Parallel task scheduling

For any poset, there is a legal parallel schedule that runs in t steps, where t is the length of the longest chain.

We will in fact prove:

Theorem

For a finite poset $(S, \preceq)$ with length of longest chain $= t$, we can partition S into t subsets $S_1, \ldots, S_t$ such that $\forall i \in \{1, \ldots, t\}$, $\forall a \in S_i$, if $b \preceq a, b \neq a$ then $b \in S_1 \cup \ldots \cup S_{i-1}$.

Assuming this theorem,

- Observe that we can schedule all of S_i at time i (since we know that all previous tasks were done earlier!).
- Thus, each S_i is an anti-chain.
- This solves the parallel task scheduling problem.

Parallel task scheduling and chains

Theorem

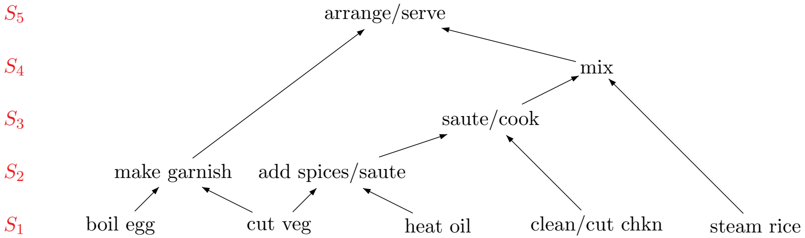
For a finite poset $(S, \preceq)$ with length of longest chain $= t$, we can partition S into t subsets $S_1, \ldots, S_t$ such that $\forall i \in \{1, \ldots, t\}$, $\forall a \in S_i$, if $b \preceq a, b \neq a$ then $b \in S_1 \cup \ldots \cup S_{i-1}$.

Parallel task scheduling and chains

Theorem

For a finite poset $(S, \preceq)$ with length of longest chain $= t$, we can partition S into t subsets $S_1, \ldots, S_t$ such that $\forall i \in \{1, \ldots, t\}$, $\forall a \in S_i$, if $b \preceq a, b \neq a$ then $b \in S_1 \cup \ldots \cup S_{i-1}$.

Proof: Put each $a \in S$ in S_i such that i is the length of the longest chain ending at a.

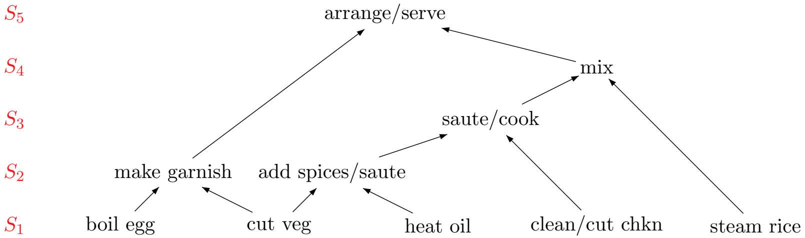


Parallel task scheduling and chains

Theorem

For a finite poset $(S, \preceq)$ with length of longest chain $= t$, we can partition S into t subsets $S_1, \ldots, S_t$ such that $\forall i \in \{1, \ldots, t\}$, $\forall a \in S_i$, if $b \preceq a, b \neq a$ then $b \in S_1 \cup \ldots \cup S_{i-1}$.

Proof: Put each $a \in S$ in S_i such that i is the length of the longest chain ending at a. Now proof by contradiction:



Parallel task scheduling and chains

Theorem

For a finite poset $(S, \preceq)$ with length of longest chain $= t$, we can partition S into t subsets $S_1, \ldots, S_t$ such that $\forall i \in \{1, \ldots, t\}$, $\forall a \in S_i$, if $b \preceq a, b \neq a$ then $b \in S_1 \cup \ldots \cup S_{i-1}$.

Proof: Put each $a \in S$ in S_i such that i is the length of the longest chain ending at a. Now proof by contradiction:

- Suppose $\exists i, a \in S_i, b \preceq a, b \neq a$ but $b \notin S_1 \cup \ldots \cup S_{i-1}$.

Parallel task scheduling and chains

Theorem

For a finite poset $(S, \preceq)$ with length of longest chain $= t$, we can partition S into t subsets $S_1, \ldots, S_t$ such that $\forall i \in \{1, \ldots, t\}$, $\forall a \in S_i$, if $b \preceq a, b \neq a$ then $b \in S_1 \cup \ldots \cup S_{i-1}$.

Proof: Put each $a \in S$ in S_i such that i is the length of the longest chain ending at a. Now proof by contradiction:

- Suppose $\exists i, a \in S_i, b \preceq a, b \neq a$ but $b \notin S_1 \cup \ldots \cup S_{i-1}$.
- By defn of S_i, $\exists$ chain of length at least i ending at b.

Parallel task scheduling and chains

Theorem

For a finite poset $(S, \preceq)$ with length of longest chain $= t$, we can partition S into t subsets $S_1, \ldots, S_t$ such that $\forall i \in \{1, \ldots, t\}$, $\forall a \in S_i$, if $b \preceq a, b \neq a$ then $b \in S_1 \cup \ldots \cup S_{i-1}$.

Proof: Put each $a \in S$ in S_i such that i is the length of the longest chain ending at a. Now proof by contradiction:

- Suppose $\exists i, a \in S_i, b \preceq a, b \neq a$ but $b \notin S_1 \cup \ldots \cup S_{i-1}$.
- By defn of S_i, $\exists$ chain of length at least i ending at b.
- But now, $b \preceq a, b \neq a$ implies we can extend the chain to chain of length $\geq i + 1$, ending at a.
- But then a cannot be in S_i. Contradiction. $\square$

Consequences for chains and anti-chains

Since each S_i was an anti-chain, a celebrated result follows...

Corollary (Mirsky's theorem, 1971)

If the longest chain in a poset $(S, \leq)$ is of length t, then S can be partitioned into t anti-chains.

Consequences for chains and anti-chains

Since each S_i was an anti-chain, a celebrated result follows...

Corollary (Mirsky's theorem, 1971)

If the longest chain in a poset $(S, \leq)$ is of length t, then S can be partitioned into t anti-chains.

(H.W) Apply this on the poset $(\mathcal{P}(S), \subseteq)$, where $S = \{1, 2, 3\}$. What does it imply for this poset?

Consequences for chains and anti-chains

Since each S_i was an anti-chain, a celebrated result follows...

Corollary (Mirsky's theorem, 1971)

If the longest chain in a poset $(S, \leq)$ is of length t, then S can be partitioned into t anti-chains.

(H.W) Apply this on the poset $(\mathcal{P}(S), \subseteq)$, where $S = \{1, 2, 3\}$. What does it imply for this poset?

Another corollary (Dilworth's Lemma)

For all $t > 0$, any poset with n elements must have

- either a chain of length greater than t
- or an antichain with at least $\frac{n}{t}$ elements.

(H.W): Prove it!