1. a. List all the binary relations on the set $\{0,1\}$.

here we are listing the graphs of the relations

$\emptyset$, $\{(0,0)\}$, $\{(0,1)\}$, $\{(1,0)\}$, $\{(1,1)\}$

$\{(0,0),(0,1)\}$, $\{(0,0),(1,0)\}$, $\{(0,0),(1,1)\}$,

$\{(0,1),(1,0)\}$, $\{(0,1),(1,1)\}$, $\{(1,1),(1,0)\}$,

$\{(0,0),(0,1),(1,0)\}$, $\{(0,0),(0,1),(1,1)\}$, $\{(0,0),(1,0),(1,1)\}$,

$\{(0,1),(1,0),(1,1)\}$,

$\{(0,0),(0,1),(1,0),(1,1)\}$

b. Over the domain $\{0,1\}$, which of these relations are equivalence relations?

Listing the relations by property on $A = \{0,1\}$

Reflexive: 8, 13, 14, 16

Symmetric: 1, 2, 5, 8, 9, 12, 15, 16

Transitive: 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 13, 14, 16

Equivalence relations are all 3, so: $\boxed{8, 16}$

Weak/strict Partial orders?

A binary relation is a partial order iff it is transitive & antisymmetric

Antisymmetric: 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 13, 14

$\Rightarrow$ The partial orders are 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 13, 14

A partial order is weak iff it is reflexive

$\boxed{\text{Weak partial orders: } 8, 13, 14}$

Strict iff irreflexive

$\boxed{\text{Strict partial orders: } 1, 3, 4}$

2. We partially order the power set, $\mathcal{P}(\{1,2,\dots,n\})$, by the subset relation, $\subseteq$.

a. Describe the maximum length chain in $\mathcal{P}(\{1,2,\dots,n\})$. Breifly explain why there can't be a longer chain than the one described.

a chain is a set of elements $\ni$ any two elements in the set are comparable

Maximum chain length is $\boxed{n+1}$, and this chain is

$\boxed{\{\emptyset, \{1\}, \{1,2\}, \dots, \{1,2,\dots,n\}\}}$

this is the longest chain, since if you add another element to it, or skip an element in one of the sets in general, you will need to remove all sets smaller than or equal to that set containing the missing element, of which there will be at least one.

2.b. Describe a topological sort of $\mathcal{P}(\{1,2,\dots,n\})$, with a brief justification that your sort is correct.

A topological sort of $\subseteq$ on $\mathcal{P}(\{1,2,\dots,n\})$ is all sets of size 0, followed by all sets of size 1, .. 2, ..., the set $\{1,2,\dots,n\}$.
The total ordering performing this topological sort is $A \sqsubseteq B \iff |A| \leq |B|$
Justification.

From the definition of a topological sort, a topological sort of the partial order $\subseteq$ on the set $\mathcal{P}(\{1,2,\dots,n\})$ is a total ordering $\sqsubseteq$, on $\mathcal{P}(\{1,2,\dots,n\}) \in$ $a \subseteq b \Rightarrow a \sqsubseteq b$, $a,b \in \mathcal{P}(\{1,2,\dots,n\})$

If $a \subseteq b$ then a must either have less elements than b, or $a=b$

Case 1 $|a|<|b|$:

from the definition of our topological sort $a \sqsubseteq b$

Case 2 $a=b$:

$|a|=|b| \Rightarrow a \sqsubseteq b$ from the def of our total ordering

So $a \subseteq b \Rightarrow a \sqsubseteq b$ $\forall a,b \in \mathcal{P}(\{1,2,\dots,n\}) \Rightarrow$ our topological sort is valid.

2.c. Use Dilworth's Lemma to show that there must be an antichain of size $\geq 2^n/(n+1)$. Describe the biggest anti-chain you can find.

Proof:

Let $t = n+1$

The partially ordered set $P(\{1,2,\dots,n\})$ has 2^n elements because it is a powerset of a set with n elements.

In 2.a. we show that the maximum chain length is $n+1$.

Dilworth's Lemma states: $\forall t$, every partially ordered set with m elements must have either a chain of size $> t$ or an antichain of size m/t

Since $t > n$, by Dilworth's lemma $\exists$ an anti-chain of size $\geq \frac{m}{t} = \left(\frac{2^n}{n+1}\right)$ //

The largest antichain I can find is the set of all subsets of $\{1,2,\dots,n\}$ of size $\lceil n/2 \rceil$ or $\lfloor n/2 \rfloor$. The case for $\lceil n/2 \rceil$ is briefly described below.

If we topologically sort $P(\{1,2,\dots,n\})$ into all sets of size 0 followed by all sets of size 1, ..., through all sets of size n, we find the DAG shown below, with anti-chains in rows (represented by "—*—"). $\lceil n/2 \rceil$ is $\geq$ the widest section of this DAG.

$$\begin{tikzcd}
\text{Size:} & & & & \text{Largest anti-chain} \arrow[dddd, bend left] \\
0 & & \emptyset \arrow[dl] \arrow[d] \arrow[dr] & & \\
1 & \{1\} \arrow[r, dash, "\ast" description] \arrow[d] & \{2\} \arrow[r, dash, "\ast" description] \arrow[d] & \{n\} \arrow[d] & \\
\vdots & \vdots \arrow[d] & \vdots \arrow[d] & \vdots \arrow[d] & \\
\lceil n/2 \rceil & \{1,2,\dots,\lceil n/2 \rceil\} \arrow[r, dash, "\ast" description] \arrow[ddr] & \{1,2,\dots,\lceil n/2 \rceil -1, \lceil n/2 \rceil +1\} \arrow[r, dash, "\ast" description] \arrow[dd] & \{\lceil n/2 \rceil, \lceil n/2 \rceil +1, \dots, n\} \arrow[ddl] & \\
\vdots & & & & \\
n & & \{1,2,\dots,n\} & &
\end{tikzcd}$$

3. Consider the natural numbers partially ordered by divisibility

a. Prove that this partial order has an infinite chain.

Proof

We will prove that the chain $\{1, 2, 4, \dots, 2^n\}$ $\forall n \in \mathbb{N}$ is a [illegible] chain by induction

Base Case: $n=0$ 1 is divisible by 1, so $\{1\}$ is a chain in $\mathbb{N}$ ordered by divisibility

Inductive step: for $n \in \mathbb{N}$ $\{1, 2, \dots, 2^n\}$ is a chain

Consider the set $\{1, 2, \dots, 2^n\} \cup 2^{n+1}$

Notice $2^{n+1} = 2(2^n)$, so 2^{n+1} is divisible by 2^n

Since the first part of the union is a chain, 2^{n+1} is also divisible by the remaining elements of $\{1, 2, \dots, 2^n\}$ since it is divisible by 2^n and divisibility is transitive (using the inductive hypothesis)

Then by induction $\{1, 2, \dots, 2^n\}$ is a chain $\forall n \in \mathbb{N}$.

$\mathbb{N}$ is an infinite set, so an infinite chain of this kind can be constructed by using $\mathbb{N}$. $\Rightarrow \exists$ an infinite chain //

3.b. Prove that this partial order has an infinite anti-chain.

Proof

Consider the set of prime numbers.

By the definition of prime, each number in the set is only divisible by 1 and itself

So each element of the set is not divisible, and incomparable by the divisibility relation.

By the definition of an anti-chain, the set of prime numbers forms an anti-chain. (Since prime $\subseteq \mathbb{N}$)

The set of prime numbers is an infinite set, so the set is an infinite anti-chain

So there exists an infinite antichain on $\mathbb{N}$ partially ordered by divisibility, namely the set of all prime numbers. //

3.c. Now restrict the domain to the natural numbers $\leq n$. Consider the chain

(1) $1 \preceq_R 2 \preceq_R 4 \preceq_R \cdots \preceq_R 2^{\lfloor \log_2 n \rfloor}$. Prove that it is maximal.

Proof:

Suppose there exists a longer chain $a_0 \preceq_R a_1 \preceq_R a_2 \preceq_R \cdots \preceq_R a_m$.

Chain (1) is of length $\lfloor \log_2 n \rfloor + 1$, so $m \geq \lfloor \log_2 n \rfloor + 1$

Since adjacent a_i's are divisible, we can let $a_i = p_i a_{i-1}$ $\forall i \in \{1, 2, \ldots, m\}$

where $p_i \geq 2$ (since it must be an integer > 1)

Then $a_m = \prod_{i=1}^{m} p_i a_0$ where $a_0 \geq 1$

So
$$\begin{aligned} a_m &= \prod_{i=1}^{m} p_i a_0 \geq 2^m a_0 \quad \text{(since } p_i \geq 2\text{)} \\ &\geq 2^m \\ &\geq 2^{\lfloor \log_2 n \rfloor + 1} \\ &> 2^{\log_2 n} \\ &= n \end{aligned}$$

$\Rightarrow\Leftarrow$ However this leads to a contradiction since $a_m > n \notin \{x \in \mathbb{N} \mid x \leq n\}$

Thus there exists no chain longer than $\lfloor \log_2 n \rfloor + 1$

So (1) is maximal //

d. Let c be the length of the power of 2 chain. By Dilworth's lemma there is an antichain of length n/c. Describe one.

$c \leq \lfloor \log_2 n \rfloor + 1$

This anti chain is $\{2^{\lfloor \log_2 n-1 \rfloor}+1, 2^{\lfloor \log_2 n-1 \rfloor}+2, \ldots, 2^{\lfloor \log_2 n \rfloor}\}$

The solution writes this simpler as $\{\lfloor \frac{n}{2} \rfloor + 1, \lfloor \frac{n}{2} \rfloor + 2, \ldots, n\}$, but they are the same set.

4. We consider DAG's where each vertex represents a task to be completed. If there is a path from one vertex, v, to another vertex, w, then the v task must be completed before the w task. Assuming all tasks take unit time to complete, we showed in the Notes that the minimum time schedule to complete all the tasks is the size (number of vertices), t, of the longest path (chain) in the DAG.

Formally, a schedule for a DAG is a partition of vertices. Each block of the partition is supposed to correspond to a set of tasks that are to be performed simultaneously. The number of processors required by a schedule is the maximum number of tasks that are scheduled to be performed simultaneously.

(a) Describe purely in terms of graph, partition, and partial order properties:

• Exactly the properties a vertex partition of a DAG must satisfy in order to represent a possible schedule for vertex tasks.

A schedule for a DAG is a partition of the edges of G into a sequence of $\overset{k}{\vee}$ anti-chains called blocks, ordered such that the sequence of the elements in B_1 in any order, followed by those of B_2 in any order, through B_k is a topological sort of the partial order on G

• Total time required to complete a schedule.

Number of blocks

• Number of processors required to complete a schedule

Number of elements within the largest block in the schedule.

4.b. Give a small example of a DAG with more than one minimum time schedule.

Consider the DAG $V = \{1, 2, 3\}$, $E = \{1 \to 2\}$

Then $\{\{1\}, \{2, 3\}\}$ and $\{\{1, 3\}, \{2\}\}$ are both minimum time schedules.

C. Explain why any schedule that requires only p processors to complete n tasks must take time at least $\lceil n/p \rceil$.

Suppose $\exists$ a schedule $\ni$ the n tasks can be completed in time $t < \lceil n/p \rceil$ on only p processors

Then the tasks must be completed in $< \lceil n/p \rceil$ steps since the steps are of unit time.

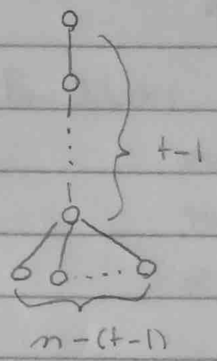
Then there must be $< \lceil n/p \rceil$ blocks in the schedule since blocks consist of tasks that can be completed simultaneously.

So $\exists$ a block of size $> p$ since $(< \lceil n/p \rceil$ blocks$)(p($blocksize$)) < n$

However the number of processors required to complete a given schedule is $=$ to the largest block size as argued in part a.

$\Rightarrow \cdot \Leftarrow$ since we only have p processors, this leads to a contradiction.

d. Let $D_{n,t}$ be the DAG with n vertices that consists of a directed path of $(t-1)$ vertices ending with edges from the final, $(t-1)$st, vertex on the path directly to each of the remaining $n-(t-1)$ vertices, as in the following figure.



What is the minimum time schedule for $D_{n,t}$? Explain why it is unique. How many processors does it require?

thm 9.11 Break the DAG into t blocks, since it is the longest chain length is $(t-1+1=t)$. Blocks $B_1, \dots, B_{t-1}$ are single tasks $1, \dots, t-1$ since they are in a single chain and must be run sequentially. The block B_t is the $n-t+1$ tasks at the bottom of the DAG run in parallel. This schedule takes time t since there are t blocks.

This is the unique minimum time schedule since the first $t-1$ tasks must run sequentially since they form a chain. After completing these tasks the longest remaining chain is of length 1 since the $n-t+1$ remaining tasks form an anti-chain $\Rightarrow$ they can be all run in parallel using $n-t+1$ processors.

f. Show that every DAG with n-vertices and maximum chain size, t, has a p-processor schedule that runs in time $M(n,t,p)$.

Hint: Induction, you decide on what variable. You may find it helpful to use the fact that if $a \geq b \geq 0$, then $\lceil a-b \rceil \leq 1 + \lceil a \rceil - \lceil b \rceil \ \forall a,b \in \mathbb{R}$ (1)

Proof by Induction on t

Induction hypothesis: $P(t)$: every DAG with $n \in \mathbb{N}$ vertices and maximum chain size t has a $p \in \mathbb{N}$ processor schedule that runs in time $M(n,t,p) = (t-1) + \lceil \frac{n-(t-1)}{p} \rceil$

Base case: $P(1)$

Consider a DAG with n vertices and a maximum chain size $t=1$ with $p \in \mathbb{N}$ processors.

The vertices are disjoint since there are no chains of length >1

The tasks can be divided into $\lceil \frac{n}{p} \rceil$ blocks of size $\leq p$, making the time schedule $\lceil \frac{n}{p} \rceil = (1-1) + \lceil \frac{n-(1-1)}{p} \rceil = M(n,1,p)$

So $P(1)$ is true, forming a basis for induction

Inductive Step: Assume $P(t)$ is true $(t \geq 1)$

Consider a DAG with $n \in \mathbb{N}$ vertices, $p \in \mathbb{N}$ processors, with a maximum chain size $t+1$.

If you take the $k \geq 1$ ends of each max length chain and form a subgraph H by removing these k vertices, we now have a DAG H with $n-k$ vertices and max chain length t.

by our induction hypothesis, H has a schedule that runs in time $M(n-k,t,p)$

We can extend H to G by adding the k endpoints back in $\lceil k/p \rceil$ disjoint (from each other) blocks. This gives us the following time schedule for G

$M(n-k,t,p) + \lceil \frac{k}{p} \rceil$ (2)

Now we want to show $(2) \leq M(n,t+1,p)$

We can rewrite (2) as $(t-1) + \lceil \frac{n-k+(t-1)}{p} \rceil + \lceil \frac{k}{p} \rceil$ from def of $M(n,t,p)$

$= (t-1) + \lceil \frac{n-t}{p} - \frac{(k-1)}{p} \rceil + \lceil \frac{k}{p} \rceil$

Case 1 ($k-1$ is not divisible by p):

Then $\left\lceil \frac{k-1}{p} \right\rceil = \left\lceil \frac{k}{p} \right\rceil$

So $(2) \leq (t-1) + 1 + \left\lceil \frac{n-t}{p} \right\rceil \cancel{- \left\lceil \frac{k-1}{p} \right\rceil + \left\lceil \frac{k}{p} \right\rceil}$ by (1)

$= (t+1) - 1 + \left\lceil \frac{n-(t+1-1)}{p} \right\rceil = M(n, t+1, p)$

Case 2 ($(k-1)$ is divisible by p):

Then $\frac{k-1}{p} \in \mathbb{Z} \Rightarrow \left\lceil \frac{k-1}{p} \right\rceil = \frac{k-1}{p}$

so $(2) = (t-1) + \left(\left\lceil \frac{n-t}{p} \right\rceil - \frac{k-1}{p} + \left(\frac{k-1}{p}\right) + 1\right.$ since $\left\lceil \frac{k}{p} \right\rceil = \frac{k-1}{p} + 1$

$= (t+1) - 1 + \left\lceil \frac{n-(t+1-1)}{p} \right\rceil = M(n, t+1, p)$

So there exists a time schedule that runs in $\leq M(n, t+1, p)$

$\Rightarrow$ There must exist a time schedule that runs in $M(n, t+1, p)$

So $P(t) \Rightarrow P(t+1)$

Thus, by induction $P(t)$ is true $\forall t \in \mathbb{N}^+$ //

(e) Describe a minimum time p-processor schedule for $D_{n,t}$. Write a simple formula for this minimum time, $M(n,t,p)$.

$$M(n,t,p) = \underset{\substack{\uparrow \\ \text{first } B_{t-1} \text{ blocks}}}{t-1} + \lceil (n-t+1)/p \rceil$$

running p of the $n-t+1$ tasks at a time, when $<p$ remain, run all those at the same time (why we round up)