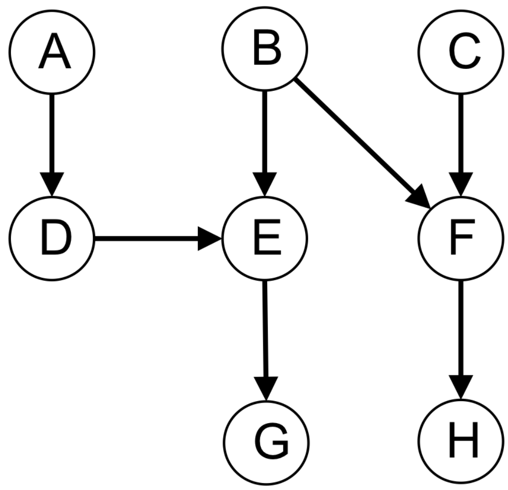
Problem 10.15.
Let $\{A, ..., H\}$ be a set of tasks that we must complete. The following DAG describes which tasks must be done before others, where there is an arrow from S to T iff S must be done before T.



(a) Write the longest chain.

(b) Write the longest antichain.

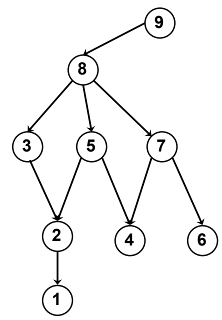
(c) If we allow parallel scheduling, and each task takes 1 minute to complete, what is the minimum amount of time needed to complete all tasks?

Problem 10.16.
Describe a sequence consisting of the integers from 1 to 10,000 in some order so that there is no increasing or decreasing subsequence of size 101.

Problem 10.17.
Suppose the vertices of a DAG represent tasks taking unit time to complete, and the edges indicate prerequisites among the tasks. Assume there is no bound on how many tasks may be performed in parallel.

What is the smallest number of vertices (tasks) possible in a DAG for which there is more than one minimum time schedule? Carefully justify your answer.

Problem 10.18.
The following DAG describes the prerequisites among tasks $\{1, \ldots, 9\}$.



(a) If each task takes **one hour** to complete, what is the minimum parallel time to complete all the tasks? Briefly explain.

(b) What is the minimum parallel time if no more than two tasks can be completed in parallel? Briefly explain.

Problem 10.19.
The following DAG describes the prerequisites among tasks $\{1, \ldots, 9\}$.

(a) If each task takes unit time to complete, what is the minimum time to complete all the tasks? Briefly explain.

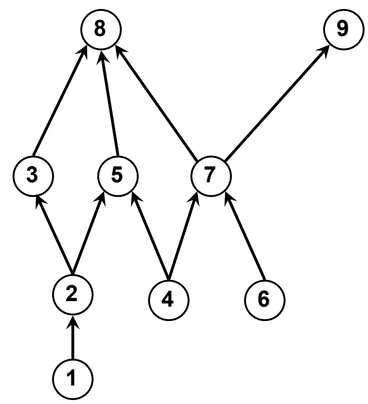
(b) What is the minimum time if no more than two tasks can be completed in parallel? Briefly explain.

Problem 10.20.
Answer the following questions about the dependency DAG shown in Figure 10.12. Assume each node is a task that takes 1 second.

(a) What is the largest chain in this DAG? If there is more than one, only give one.

(b) What is the largest antichain? (Again, give only one if you find there are more than one). Prove there isn't a larger antichain.

(c) How much time would be required to complete all the tasks with a single processor?



(d) How much time would be required to complete all the tasks if there are unlimited processors available.

(e) What is the smallest number of processors that would still allow completion of all the tasks in optimal time? Show a schedule proving it.

Class Problems

Problem 10.21.

The table below lists some prerequisite information for some subjects in the MIT Computer Science program (in 2006). This defines an indirect prerequisite relation that is a DAG with these subjects as vertices.

$$\begin{array}{rr}
18.01 \to 6.042 & 18.01 \to 18.02 \\
18.01 \to 18.03 & 6.046 \to 6.840 \\
8.01 \to 8.02 & 6.001 \to 6.034 \\
6.042 \to 6.046 & 18.03, 8.02 \to 6.002 \\
6.001, 6.002 \to 6.003 & 6.001, 6.002 \to 6.004 \\
6.004 \to 6.033 & 6.033 \to 6.857
\end{array}$$

(a) Explain why exactly six terms are required to finish all these subjects, if you can take as many subjects as you want per term. Using a *greedy* subject selection

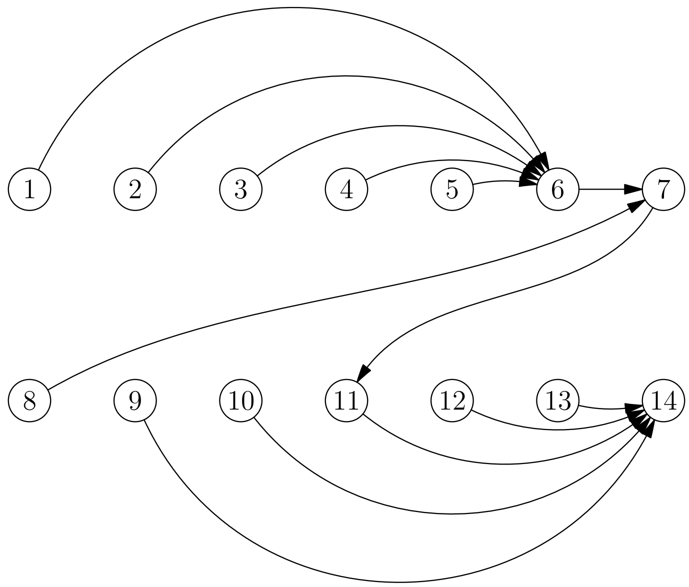


Figure 10.12 Task DAG

strategy, you should take as many subjects as possible each term. Exhibit your complete class schedule each term using a greedy strategy.

(b) In the second term of the greedy schedule, you took five subjects including 18.03. Identify a set of five subjects not including 18.03 such that it would be possible to take them in any one term (using some nongreedy schedule). Can you figure out how many such sets there are?

(c) Exhibit a schedule for taking all the courses—but only one per term.

(d) Suppose that you want to take all of the subjects, but can handle only two per term. Exactly how many terms are required to graduate? Explain why.

(e) What if you could take three subjects per term?

Problem 10.22.
A pair of Math for Computer Science Teaching Assistants, Lisa and Annie, have decided to devote some of their spare time this term to establishing dominion over the entire galaxy. Recognizing this as an ambitious project, they worked out the following table of tasks on the back of Annie’s copy of the lecture notes.

1. **Devise a logo** and cool imperial theme music - 8 days.
2. **Build a fleet** of Hyperwarp Stardestroyers out of eating paraphernalia swiped from Lobdell - 18 days.
3. **Seize control** of the United Nations - 9 days, after task #1.
4. **Get shots** for Lisa’s cat, Tailspin - 11 days, after task #1.
5. **Open a Starbucks chain** for the army to get their caffeine - 10 days, after task #3.
6. **Train an army** of elite interstellar warriors by dragging people to see *The Phantom Menace* dozens of times - 4 days, after tasks #3, #4, and #5.
7. **Launch the fleet** of Stardestroyers, crush all sentient alien species, and establish a Galactic Empire - 6 days, after tasks #2 and #6.
8. **Defeat Microsoft** - 8 days, after tasks #2 and #6.

We picture this information in Figure 10.13 below by drawing a point for each task, and labelling it with the name and weight of the task. An edge between two points indicates that the task for the higher point must be completed before beginning the task for the lower one.

(a) Give some valid order in which the tasks might be completed.

Lisa and Annie want to complete all these tasks in the shortest possible time. However, they have agreed on some constraining work rules.

- Only one person can be assigned to a particular task; they cannot work together on a single task.
- Once a person is assigned to a task, that person must work exclusively on the assignment until it is completed. So, for example, Lisa cannot work on building a fleet for a few days, run to get shots for Tailspin, and then return to building the fleet.

(b) Lisa and Annie want to know how long conquering the galaxy will take. Annie suggests dividing the total number of days of work by the number of workers, which is two. What lower bound on the time to conquer the galaxy does this give, and why might the actual time required be greater?

(c) Lisa proposes a different method for determining the duration of their project. She suggests looking at the duration of the *critical path*, the most time-consuming sequence of tasks such that each depends on the one before. What lower bound does this give, and why might it also be too low?

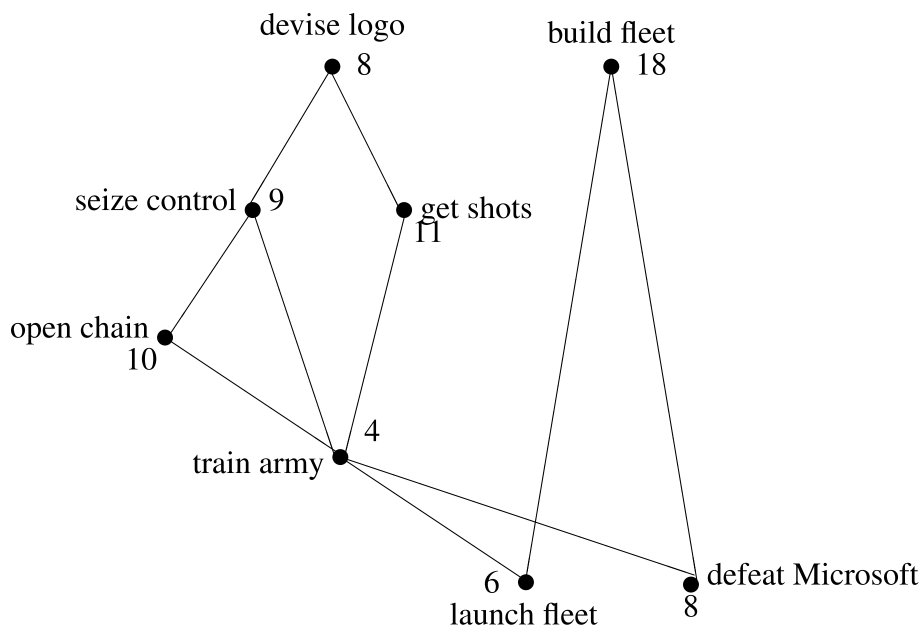


Figure 10.13 Graph representing the task precedence constraints.

(d) What is the minimum number of days that Lisa and Annie need to conquer the galaxy? No proof is required.

Problem 10.23.
Answer the following questions about the powerset pow($\{1, 2, 3, 4\}$) partially ordered by the strict subset relation $\subset$.

(a) Give an example of a maximum length chain.

(b) Give an example of an antchain of size 6.

(c) Describe an example of a topological sort of pow($\{1, 2, 3, 4\}$).

(d) Suppose the partial order describes scheduling constraints on 16 tasks. That is, if

$$A \subset B \subseteq \{1, 2, 3, 4\},$$

then A has to be completed before B starts.[16] What is the minimum number of processors needed to complete all the tasks in minimum parallel time?

[16] As usual, we assume each task requires one time unit to complete.

Prove it.

(e) What is the length of a minimum time **3**-processor schedule?

Prove it.

Homework Problems

Problem 10.24.
The following operations can be applied to any digraph, G:

1. Delete an edge that is in a cycle.
2. Delete edge $\langle u \to v \rangle$ if there is a path from vertex u to vertex v that does not include $\langle u \to v \rangle$.
3. Add edge $\langle u \to v \rangle$ if there is no path in either direction between vertex u and vertex v.

The procedure of repeating these operations until none of them are applicable can be modeled as a state machine. The start state is G, and the states are all possible digraphs with the same vertices as G.

(a) Let G be the graph with vertices $\{1, 2, 3, 4\}$ and edges

$$\{\langle 1 \to 2 \rangle, \langle 2 \to 3 \rangle, \langle 3 \to 4 \rangle, \langle 3 \to 2 \rangle, \langle 1 \to 4 \rangle\}$$

What are the possible final states reachable from G?

A *line graph* is a graph whose edges are all on one path. All the final graphs in part (a) are line graphs.

(b) Prove that if the procedure terminates with a digraph H then H is a line graph with the same vertices as G.

Hint: Show that if H is *not* a line graph, then some operation must be applicable.

(c) Prove that being a DAG is a preserved invariant of the procedure.

(d) Prove that if G is a DAG and the procedure terminates, then the walk relation of the final line graph is a topological sort of G.

Hint: Verify that the predicate

$$P(u, v) ::= \text{there is a directed path from } u \text{ to } v$$

is a preserved invariant of the procedure, for any two vertices u, v of a DAG.

(e) Prove that if G is finite, then the procedure terminates.

Hint: Let s be the number of cycles, e be the number of edges, and p be the number of pairs of vertices with a directed path (in either direction) between them. Note that $p \le n^2$ where n is the number of vertices of G. Find coefficients a, b, c such that $as + bp + e + c$ is nonnegative integer valued and decreases at each transition.

Problem 10.25.
Let $\prec$ be a strict partial order on a set A and let

$$A_k ::= \{a \mid \text{depth}(a) = k\}$$

where $k \in \mathbb{N}$.

(a) Prove that $A_0, A_1, \ldots$ is a parallel schedule for $\prec$ according to Definition 10.5.7.

(b) Prove that A_k is an antichain.

Problem 10.26.
We want to schedule n tasks with prerequisite constraints among the tasks defined by a DAG.

(a) Explain why any schedule that requires only p processors must take time at least $\lceil n/p \rceil$.

(b) Let $D_{n,t}$ be the DAG with n elements that consists of a chain of $t-1$ elements, with the bottom element in the chain being a prerequisite of all the remaining elements 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?

(c) Write a simple formula $M(n, t, p)$ for the minimum time of a p-processor schedule to complete $D_{n,t}$.

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

Hint: Use induction on t.