

Math for CS 2015/2019 solutions to “In-Class Problems Week 7, Fri. (Session 17)”

<https://github.com/spamegg1>

October 30, 2022

Contents

1	Problem 1	2
1.1	(a)	2
1.2	(b)	3
1.3	(c)	3
1.4	(d)	4
1.5	(e)	4
2	Problem 2	4
2.1	(a)	5
2.2	(b)	6
2.3	(c)	6
2.4	(d)	7
3	Problem 3	7
3.1	(a)	7
3.2	(b)	7
3.3	(c)	8
4	Problem 4	8
4.1	(a)	8
4.2	(b)	9
4.3	(c)	9
4.4	(d)	10
4.5	(e)	10
4.6	(f)	10

1 Problem 1

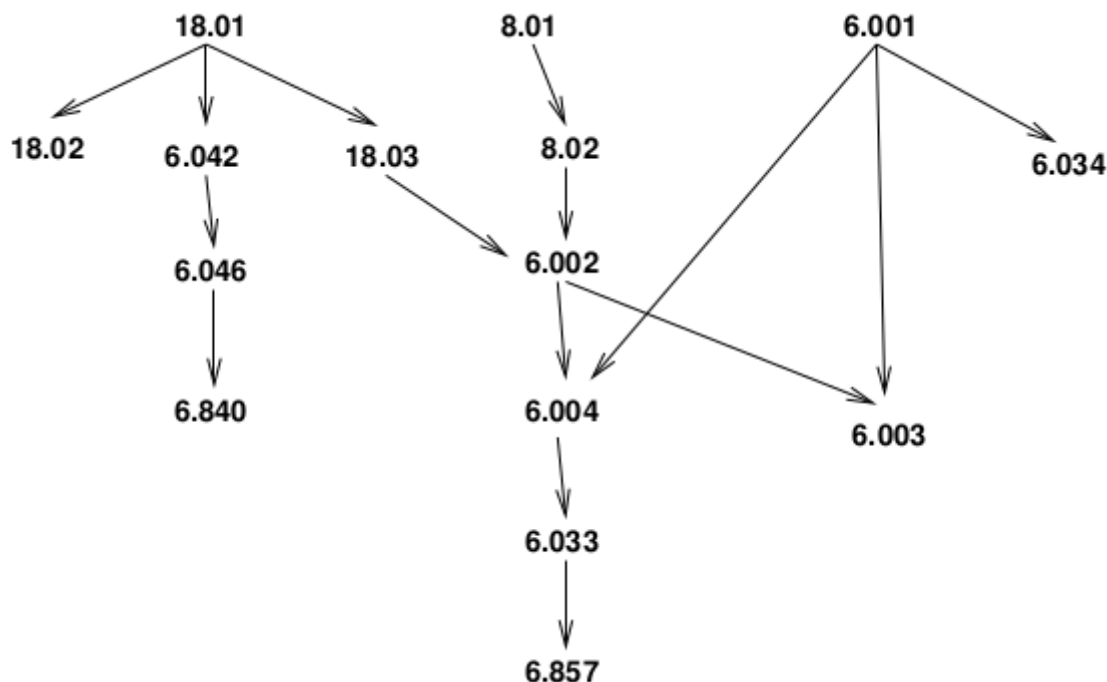
The table below lists some prerequisite information for some subjects in the MIT Computer Science program (in 2006). This defines an indirect prerequisite relation that is a DAG with these subjects as vertices.

$18.01 \rightarrow 6.042$	$18.01 \rightarrow 18.02$
$18.01 \rightarrow 18.03$	$6.046 \rightarrow 6.840$
$8.01 \rightarrow 8.02$	$6.001 \rightarrow 6.034$
$6.042 \rightarrow 6.046$	$18.03, 8.02 \rightarrow 6.002$
$6.001, 6.002 \rightarrow 6.003$	$6.001, 6.002 \rightarrow 6.004$
$6.004 \rightarrow 6.033$	$6.033 \rightarrow 6.857$

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

Proof. It helps to have a diagram of the direct prerequisite relation:



There is a chain of length 6:

$$8.01 \rightarrow 8.02 \rightarrow 6.002 \rightarrow 6.004 \rightarrow 6.033 \rightarrow 6.857$$

So we need at least 6 terms.

So six terms are necessary, because at most one of these subjects can be taken each term.

There is no longer chain than length 6, so with the greedy strategy you will take six terms. Here are the subjects you take in successive terms.

Here's a greedy schedule:

Term 1: take 6.001, 8.01, 18.01,

Term 2: take 6.034, 6.042, 8.02, 18.02, 18.03,

Term 3: take 6.002, 6.046,

Term 4: take 6.003, 6.004, 6.840,

Term 5: take 6.033,

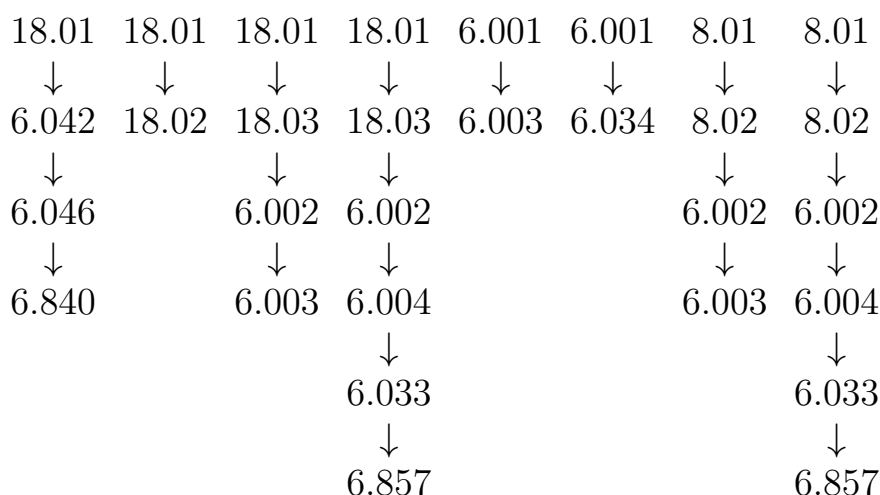
Term 6: take 6.857.

□

1.2 (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?

Proof. Here are all the chains:



We're looking for an antichain that does not include 18.03. Every such antichain will have to include 18.02, 6.003, 6.034. Then a fourth subject could be any of 6.042, 6.046, and 6.840. The fifth subject could then be any of 6.004, 6.033, and 6.857. This gives a total of nine antichains of five subjects. □

1.3 (c)

Exhibit a schedule for taking all the courses - but only one per term.

Proof. We're asking for a topological sort. There are many. One is:

18.01, 8.01, 6.001, 18.02, 6.042, 18.03, 8.02, 6.034, 6.046, 6.002, 6.840, 6.004, 6.003, 6.033, 6.857. □

1.4 (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.

Proof. There are $\lceil 15/2 \rceil = 8$ terms necessary. The schedule below shows that 8 terms are sufficient as well:

1:	18.01	8.01
2:	6.001	18.02
3:	6.042	18.03
4:	8.02	6.034
5:	6.046	6.002
6:	6.840	6.004
7:	6.003	6.033
8:	6.857	

□

1.5 (e)

What if you could take three subjects per term?

Proof. From part (a) we know six terms are required even if there is no limit on the number of subjects per term. Six terms are also sufficient, as the following schedule shows:

1:	18.01	8.01	6.001
2:	6.042	18.03	8.02
3:	18.02	6.046	6.002
4:	6.004	6.003	6.034
5:	6.840	6.033	
6:	6.857		

□

2 Problem 2

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 1 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.

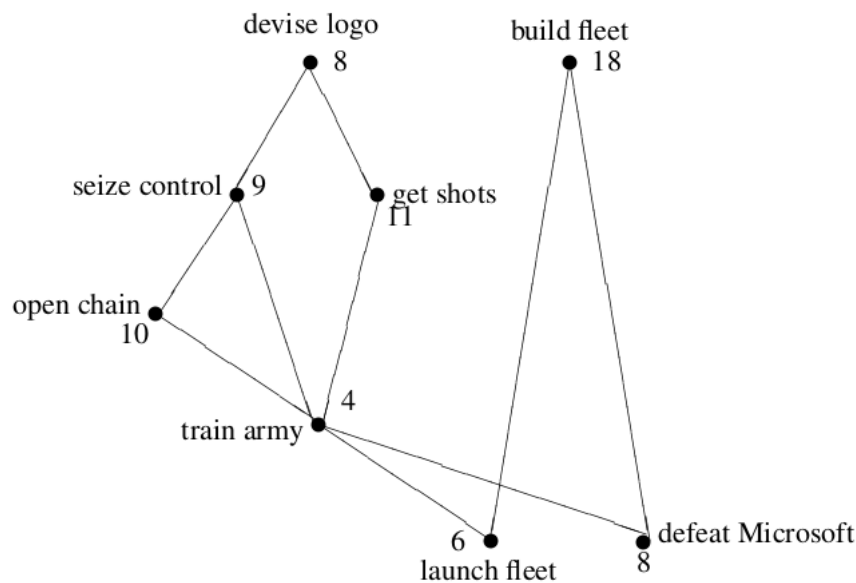


Figure 1 Graph representing the task precedence constraints.

2.1 (a)

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

Proof. We can easily find several of them. The most natural one is valid, too: #1, #2, #3, #4, #5, #6, #7, #8. □

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.

2.2 (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?

Proof.

$$\frac{8 + 18 + 9 + 11 + 10 + 4 + 6 + 8}{2} = 37 \text{ days}$$

If working together and interrupting work on a task were permitted, then this answer would be correct. However, the rules may prevent Lisa and Annie from both working all the time. For example, suppose the only task was building the fleet. It will take 18 days, not 18/2 days, to complete, because only one person can work on it and the other must sit idle. \square

2.3 (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?

Proof. The longest sequence of tasks is devising a logo (8 days), seizing the U.N. (9 days), opening a Starbucks (10 days), training the army (4 days), and then defeating Microsoft (8 days). Since these tasks must be done sequentially, galactic conquest will require at least 39 days.

If there were enough workers, this answer would be correct; however, with only two workers, Lisa and Annie may be unable to make progress on the critical path every day. For example, suppose there were only four tasks: devise logo, build fleet, seize control, get shots. Now the critical path consists of two critical tasks: devise logo, get shots, which take 19 days. But to get through this path in 19 days, some worker must be working on a critical task at all times for the 19 days. This leaves only one worker free to complete building the fleet and seizing control, which will take at least 27 days. So in fact, 27 days is the minimum time for two workers to complete these four tasks. \square

2.4 (d)

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

Proof. 40 days. Tasks could be divided as follows:

Annie: #1 (days 1-8), #3 (days 9-17), #4 (days 18-28), #8 (days 33-40).

Lisa: #2 (days 1-18), #5 (days 19-28), #6 (days 29-32), #7 (days 33-38).

It takes some care to verify that 40 days is the best you can do. If someone comes up with a simple proof of this, tell the course staff. \square

3 Problem 3

Sauron finds that conquering Middle Earth breaks down into a bunch of tasks. Each task can be completed by a horrible creature called a Ringwraith in exactly one week. Sauron realizes the prerequisite structure among the tasks defines a DAG. He has n tasks in his DAG, with a maximum length chain of t tasks.

3.1 (a)

Sauron is trying to describe various features of his scheduling problem using standard terminology. For each feature below, indicate the number of the corresponding term.

Standard Terminology

- | | |
|----------------------------------|-----------------------------------|
| 1. Indirect prerequisite | 2. Topological sort |
| 3. Chain | 4. Antichain |
| 5. Size of the largest antichain | 6. Size of the smallest antichain |
| 7. Length of the longest chain | 8. Length of the shortest chain |

1. A set of tasks that can be worked on simultaneously.
2. A possible order in which all the tasks could be completed, if only one Ringwraith were available.
3. The minimum number of weeks required to complete all tasks, if an unlimited number of Ringwraiths were available.

Proof. 1. (4) An antichain. 2. (2) Topological sort. 3. (7) Length of the longest chain. \square

3.2 (b)

If Sauron is lucky, he will be able to get away with a small crew of Ringwraiths. Write a simple formula involving n and t for the smallest number of Ringwraiths that could possibly be able to complete all n tasks in t weeks. (Do not make any additional assumptions about the relative sizes of n and t besides $t \leq n$.) Given any n and t , describe a DAG that can be completed in t weeks using this number of Ringwraiths.

Proof. If $t = n$ and the whole DAG is a single chain, then only 1 Ringwraith is needed, since no tasks can be parallelized.

More generally, for any $1 \leq t \leq n$, the smallest number of Ringwraiths needed corresponds to the situation of minimum parallelizability; in other words, having as long chains as possible.

If the DAG consists entirely of exactly max length (t) chains (maybe with the exception of the last one, which might end up shorter than t if n is not a multiple of t), then there are $\left\lceil \frac{n}{t} \right\rceil$ chains, each of which requires 1 Ringwraith, so the smallest number of Ringwraiths is $\left\lceil \frac{n}{t} \right\rceil$. □

3.3 (c)

On the other hand, if Sauron is unlucky, he may need a large crew of Ringwraiths in order to conquer Middle Earth in t weeks. Write a simple formula involving n and t for the largest number of Ringwraiths that Sauron would ever need in order to be sure of completing all n tasks in t weeks - no matter how unlucky he was. Given any n and t , describe a DAG that can be completed in t weeks and requires this number of Ringwraiths.

Proof. Since there is a maximum length chain of t tasks, there is AT LEAST one chain of length t . Having more such chains reduces the number of Ringwraiths as in part (b), so the worst case is when all the other remaining $n - t$ tasks form an antichain.

Consider the DAG of n tasks, where t of these tasks form a prerequisite chain, and the remaining $n - t$ tasks are incomparable to each other (each single task is a length-1 chain by itself) and to the chain (i.e. an antichain). Then the length t chain requires 1 Ringwraith, and the other $n - t$ tasks can be parallelized, each requiring 1 Ringwraith, for a total of $n - t + 1$ Ringwraiths. □

4 Problem 4

If a and b are distinct nodes of a digraph, then a is said to *cover* b if there is an edge from a to b and every path from a to b includes this edge. If a covers b , the edge from a to b is called a *covering edge*.

4.1 (a)

What are the covering edges in the DAG in Figure 2?

Proof. $1 \rightarrow 2$ is a covering edge, because it's the only path from 1 to 2.

$1 \rightarrow 3$ is a covering edge, because it's the only path from 1 to 3.

$1 \rightarrow 4$ is not a covering edge, because there is another path $1 \rightarrow 2 \rightarrow 4$ from 1 to 4 which doesn't include it.

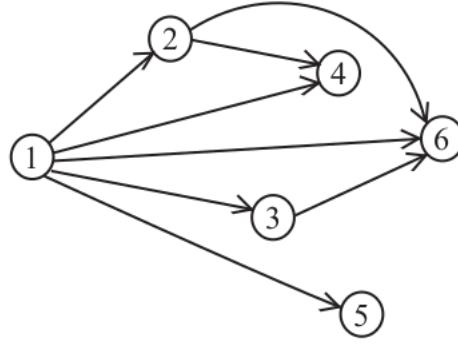


Figure 2 DAG with edges not needed in paths

$1 \rightarrow 5$ is a covering edge, because it's the only path from 1 to 5.

$1 \rightarrow 6$ is not a covering edge, because there is another path $1 \rightarrow 3 \rightarrow 6$ from 1 to 6 which doesn't include it.

Similarly $2 \rightarrow 4$, $2 \rightarrow 6$ and $3 \rightarrow 6$ are all covering edges. □

4.2 (b)

Suppose D is a finite DAG. Let $\text{covering}(D)$ be the subgraph of D consisting of only the covering edges. Explain why $\text{covering}(D)$ has the same positive walk relation as D .

Hint: Consider longest paths between a pair of vertices.

Proof. 1. We need to show that if there is a path in D between vertices $a \neq b$, then there is a path consisting only of covering edges from a to b . (This will prove that D and (D) have the same positive walk relation.)

2. Since D is a finite DAG, there must be a longest path P from a to b . (The path length is bounded above by the number of edges in D .)

3. Now every edge on this path P must be a covering edge or it could be replaced by a path of length 2 or more, yielding a longer path from a to b . □

4.3 (c)

Show that if two DAG's have the same positive walk relation, then they have the same set of covering edges.

Proof. 1. Suppose C and D are DAG's with the same positive walk relation and that $a \rightarrow b$ is a covering edge of C .

2. We want to show that $a \rightarrow b$ must also be a covering edge of D .

3. Since $a \rightarrow b$ itself defines a (length 1) positive length path in C , there must be a positive length path P in D from a to b .

4. If P has length greater than 1, then P must consist of a positive length path P_1 from a to c , followed by a positive length path P_2 from c to b , for some vertex, c .
5. Also, since D is a DAG, c cannot be a or b .
6. By (4) and (5) there must also be positive length paths Q_1 in C from a to c and Q_2 from c to b , and neither of these paths can traverse $a \rightarrow b$ or there would be a cycle.
7. Hence the path $Q_1 + Q_2$ from a to c to b is a path in C that does not traverse $a \rightarrow b$, contradicting the fact that $a \rightarrow b$ is a covering edge of C .
8. In summary, there is a length 1 path from a to b in D , namely $a \rightarrow b$, and this is the only path from a to b in D , which proves that $a \rightarrow b$ is a covering edge in D . \square

4.4 (d)

Conclude that $\text{covering}(D)$ is the unique DAG with the smallest number of edges among all digraphs with the same positive walk relation as D .

Proof. 1. By part (c), any DAG with the same positive walk relation as D must contain all the edges of $\text{covering}(D)$.

2. By part (b), $\text{covering}(D)$ has this same positive walk relation. It follows immediately that $\text{covering}(D)$ is the unique minimum-size DAG with the same positive path relation as D . \square

The following examples show that the above results don't work in general for digraphs with cycles.

4.5 (e)

Describe two graphs with vertices $\{1, 2\}$ which have the same set of covering edges, but not the same positive walk relation. (Hint: Self-loops.)

Proof. 1. Let one graph G have edges $\{(1, 2), (1, 1)\}$ and the other graph H have edges $\{(1, 2), (2, 2)\}$.

2. G and H have the same set of covering edges, namely, $(1, 2)$.

3. But in H there is a positive length path from 2 to 2, namely a path of length 1, but there is no positive length path from 2 to 2 in G . \square

4.6 (f)

(i) The *complete digraph* without self-loops on vertices 1, 2, 3 has edges between every two distinct vertices. What are its covering edges?

(ii) What are the covering edges of the graph with vertices 1, 2, 3 and edges $\langle 1 \rightarrow 2 \rangle, \langle 2 \rightarrow 3 \rangle, \langle 3 \rightarrow 1 \rangle$?

(iii) What about their positive walk relations?

Proof. (i) There are no covering edges, since there is always a length 2 path from a to b that does not use the edge $a \rightarrow b$.

(ii) All three edges are the covering edges.

(iii) They have the same positive walk relation, namely, each vertex is connected to all the vertices, including itself, by positive length paths. \square