NOTES FOR DISCRETE MATHEMATICS 37181
UTS SPRING 2019

MURRAY ELDER

ABSTRACT. These notes cover the basics of discrete mathematics: logic, mathematical proofs, set theory (mathematical notation), basic number theory, graph theory, counting, complexity and correctness of computer programs, and some other topics that might be useful.

The learning in this course will be driven by *doing problems* yourself, by doing the homework sheets, in teams during the (whiteboard) workshops, quizzes and midterm test. The lectures and these notes present the main ideas and theories, which you will master by doing lots of exercises and solving problems. You are also learning *how to write* and communicate mathematics – think of it as a new language. Set theory gives us a lot of the formalism to be able to communicate succinctly and accurately, and logic/proof methods give us a common ground for what is true/false/outside of our ability to decide.

In progress

CONTENTS

Date: Last updated: July 31, 2019.

0. Preamble: sudoku

If you know how to solve sudoku, you already know how to deduce true statements in mathematics. *Prove* (to yourself or to a friend) where the 3 in the middle 3×3 square should go in this picture:

The sudoku is a good starting point for how to think mathematically and algorithmically: what are the extreme cases, for example, a blank grid is not a very challenging sudoku, so what is the smallest number of squares that need to be filled to start (so that the solution is *unique*=only way way to fill it)? How many different sudoku puzzles (starting configurations) are there? Its a *finite* number, so can't we get a computer to run through all possibilities and count them? How much time/TB of memory would that take? How can we quantify how hard that problem is, without talking about a specific computer?

Part 1. Logic, proof, induction

We will see various *proofs* of statements in the course, so what makes a convincing, rigorous argument in mathematics and computer science?

We start by defining *truth values* for statements, and how to build up more complex statements from simple ones.

1. Introduction to formal logic and proofs

A *statement* is a sentence that can (theoretically) be assigned a value of *true* or *false*.

1. Um, like, whatever
2. All positive integers are prime
3. There is a number that is larger than π and smaller than $\sqrt{2}$
4. QUT is in Brisbane
5. In the year 4000BC, at this exact location, it was raining on the 5th of March at 10am

1

We can build up more complicated statements out of simpler ones using *logical connectives* like *and* and *or*.

If p, q are statements, then when is "p and q" true or false?

p	q	$p \wedge q$
1	1	1
1	0	0
0	1	0
0	0	0

Here $\wedge$ means *and*, 1 means *true* and 0 means *false*.

Example 1.1. $p =$"QUT is in Brisbane", $q =$"RMIT is in Geelong". Then $p \wedge q$ is false because q is false.

Try to follow this idea to complete the *truth table* for "or" which we denote by $\vee$:

p	q	$p \vee q$
1	1	
1	0	
0	1	
0	0	

If p is true then "*not* p" is false, and vice versa. We express this in a table as

p	$\neg p$
1	0
0	1

We can use truth tables to decide the truth values of more complicated statements, like $\neg p \vee q$:

p	q	$\neg p$	$\vee$	q
1	1	0	1	
1	0	0	0	
0	1	1	1	
0	0	1	1	
		(1)	(2)	

The order in which we compute columns (according to *order* of logical operations) is shown by the (1), (2).

[1] 2,3,4,5 are all statements, even if we can't ever know if 5 is true or false, it is still something that will be either true or false.

Note that this is different to saying $\neg(p \vee q)$, since the truth values are not the same:

p	q	$\neg$	$(p \vee q)$
1	1		
1	0		
0	1		
0	0		
		(2)	(1)

When two (compound) statements have the same truth values we say they are *logically equivalent*.

Exercise 1.2. Check that $\neg(p \vee q)$ is logically equivalent to $\neg p \wedge \neg q$ by showing the final column of their truth tables are the same.

p	q	$\neg p \wedge \neg q$
1	1	
1	0	
0	1	
0	0	

In mathematics and logic we have a very specific meaning for "*p implies q*", or "*if p then q*", notation $p \to q$. We define it using the following table:

p	q	$p \to q$
1	1	1
1	0	0
0	1	1
0	0	1

You may think that in English, "*if it is raining then I get wet*" means that the rain *caused* me to get wet. But in mathematics *if-then* has the meaning defined above: if "*I am wet*" is true and "*it is raining*" is false, the implication is still true. (I could be at a swimming pool).

Notice that the statement "*if p then q*" is logically equivalent to $\neg p \vee q$ "*not p, or q*".

Finally, we use the symbol $\leftrightarrow$ for "if and only if":

p	q	$p \leftrightarrow q$
1	1	1
1	0	0
0	1	0
0	0	1

I get wet **if and only if** it is raining (not when I am swimming, taking a shower, or otherwise).

Exercise 1.3. Complete these truth tables:

p	q	$(p \to \neg q) \vee q$

p	q	r	$\neg(p \to (q \wedge r))$

p	q	$\neg(p \vee q)$

p	q	$\neg p \wedge \neg q$

p	q	$\neg q \to \neg p$

p	$\neg(\neg p)$

(I am not (not happy)).

A statement that is true for all truth value assignments is called a *tautology*. For example $p \vee \neg p$ has a truth table with all 1's.

Exercise 1.4. Show that these statements are tautologies:

1. $((p \to q) \wedge p) \to q$
2. $((p \to q) \wedge \neg q) \to \neg p$
3. $\neg(p \wedge q) \leftrightarrow \neg p \vee \neg q$
4. $((p \to q) \wedge (q \to r)) \to (p \to r)$
5. $(p \to q) \leftrightarrow (\neg q \to \neg p)$

Items 1 and 2 of this exercise have Latin names: *modus ponens* and *modus tollens*. Item 3 is one of *De Morgan's laws* (the other one was Exercise 1.2) and item 4 is an argument form is called *syllogism*. In item 5, $\neg q \to \neg p$ is called the *contrapositive* of $p \to q$.

Note that $q \to p$ (called the *converse*) is **not** logically equivalent to $p \to q$. (Check the truth tables).

In Humanities/Law you might see tautological implications written in this form.

$$\begin{array}{l} p \to q \\ p \\ \hline q \end{array} \qquad \begin{array}{l} p \to q \\ \neg q \\ \hline \neg p \end{array} \qquad \begin{array}{l} p \to q \\ q \to r \\ \hline p \to r \end{array}$$

Example 1.5. From Wikipedia[3]:

> If I am an axe murderer, then I can use an axe.
> I cannot use an axe.
> Therefore, I am not an axe murderer.

Which style of argument is this? (Write it in symbols).

Let F be a statement that is always false (has truth table 0, for example, $F = q \wedge \neg q$). Then the statement

$$(\neg p \to F) \to p$$

is a tautology. It says, if not p implies something that is false, then it must be p (is true). This argument form is known as *proof by contradiction*, see below.

2. Variables and quantifiers

Statements can contain *variables*.

Example 2.1. $P(x)$ could be the statement: "the number x is greater than or equal to 3". $Q(x)$ could be the statement: "x lives in Queensland".

We refer to the *universe of discourse* to be the set of objects over which the statement could be defined, so for $P(x)$ the universe of discourse is numbers (maybe $\mathbb{R}$ or $\mathbb{Z}$ or $\mathbb{N}$, we would need to be told) and for $Q(x)$ the universe might be all people, or all students at QUT.

We have the symbols $\forall$ ="for all" and $\exists$ ="there exists".

Exercise 2.2. $\forall x \in \mathbb{Z}, x^2 > x$ reads as "for all integers x, x^2 is greater than x. [2]

Is this true?

$\exists x \in \mathbb{Z}, x^2 \leqslant x$ reads as "there exists (there is) some integer x whose square is smaller than or equal to itself.

Is this true?

Example 2.3. "All UTS students live in Bondi". We can rephrase this in English as: For all students at UTS, the student lives in Bondi. We let the universe of discourse be the set of all students at UTS. Then

$$\forall x \text{ (in the universe of discourse) } B(x)$$

where $B(x)$ is the proposition that x lives in Bondi.

Clearly this is not true, so the negation of this statement is true. Formally, to negate a quantified statement you switch $\forall$ and $\exists$ at the front, then negate the proposition.

$$\neg\left(\forall x B(x)\right) = \exists x \neg B(x)$$

There is (at least one) student at UTS who does not live in Bondi.

Exercise 2.4. Let $C(x)$ be the proposition that x lives in Cabramatta. Write the following sentence in symbols (as a quantified statement):

"All UTS students either live in Cabramatta or do not live in Bondi."

Now negate this sentence (in symbols first, then translate into English). Recall that in logic *or* includes both.

Exercise 2.5. Recall modus ponens/modus tollens: if someone lives in Bondi then they do not live in Cabramatta. x lives in Bondi. Therefore ... y lives in Cabramatta. Therefore ...

[2] $x \in \mathbb{Z}$ is some *set theory* notation, coming up. The symbol $\in$ means "in the set".

Exercise 2.6. Decide which is true or false where the universe of discourse is the real numbers:

1. $\forall x \exists y (xy = 1)$
2. $\exists y \forall x (xy = 1)$
3. $\forall x \forall y (x^2 + y^2 = x + y)$
4. $\forall x \exists y (x^2 + y^2 = x + y)$

3. P=NP

3-SAT is the following problem: on input an expression of the form

$$(x_1 \vee y_1 \vee z_1) \wedge (x_2 \vee y_2 \vee z_2) \wedge \ldots (x_n \vee y_n \vee z_n)$$

where x_i, y_i, z_i are propositions p or $\neg p$, answer yes or no: there is some assignment of truth values to the variables which makes the whole statement true.

For example

$$(p \vee q \vee \neg r) \wedge (\neg p \vee q \vee r) \wedge (p \vee \neg q \vee \neg p)$$

If I tell you a particular truth assignment, like $p = 0, q = 1, r = 0$ etc, you can easily compute (in a number of steps polynomial in n) the truth value of the statement. If an instance of a solution can be *verified* in polynomial time (number of steps), we say a problem is in NP.

If a solution can be found in polynomial time (number of steps), we say the problem is in P. No-one knows if you can find a truth assignment, or show there is none, making a general 3-SAT expression true, in polynomially many steps. If you can, you will get USD1M from the Clay Institute.

3-SAT is an important problem, even though it may seem abstract and useless, because Cook and Levin showed that every other candidate to solve the P=NP problem is related to this one. More details see for example [2].

4. Proofs

Proofs in mathematics or computer science are based on the argument forms we have seen above. You need to establish that each simple statement p, q, r is true, and then you put your statements together using a formal argument such as $(p \to q) \wedge (q \to r)$ implies $p \to r$. The formal logical structure is often hidden when we write out proofs in English. To start with, the main types of proof styles are: direct,

contrapositive, contradiction, and induction. If you do more math or theoretical computer science you will see more styles. [3]

4.1. **Direct.** Sometimes it is easy to show step-by-step that p implies q (or using syllogism, $p \to r$ and $r \to s$ and $s \to t$ and $t \to q$).

Recall that an integer is *even* if it can be written as $2d$ for some $d \in \mathbb{Z}$.

Lemma 4.1. *Let $n \in \mathbb{Z}$. If n is even then n^2 is even.*

Proof. By hypothesis, $n = 2s$ for some $s \in \mathbb{Z}$. Then $n^2 = (2s)^2 = 4s^2 = 2(2s^2)$ is even. □

Lemma 4.2. *Let a, b be non-negative real numbers. If $\sqrt{a} < \sqrt{b}$ then $a < b$.*

Proof. First note that if $x \geqslant 0$ (*i.e.* x is non-negative) then $\sqrt{x}$ is defined and is $\geqslant 0$, so we have $\sqrt{a}, \sqrt{b} \geqslant 0$.

$$\begin{array}{rcll} a & = & \sqrt{a}\sqrt{a} & \text{(by definition of } \sqrt{\cdot}\text{)} \\ & < & \sqrt{a}\sqrt{b} & \text{(since } \sqrt{a} < \sqrt{b} \text{ and } \sqrt{a} > 0\text{)} \\ & < & \sqrt{b}\sqrt{b} & \text{(since } \sqrt{a} < \sqrt{b} \text{ and } \sqrt{b} > 0\text{)} \\ & = & b & \text{(by definition of } \sqrt{\cdot}\text{).} \end{array}$$

□

4.2. **Contrapositive.** Recall that $p \to q$ has the same truth values as $\neg q \to \neg p$.

Lemma 4.3. *Let $n \in \mathbb{Z}$. If n^2 is even then n is even.*

Instead of trying to prove this directly, we will prove $\neg$ (n is even) implies $\neg$ (n^2 is even). In other words, if n is odd then n^2 is odd.

Proof. If n is odd, then $n = 2p+1$ for some $p \in \mathbb{Z}$, so $n^2 = 4p^2+4p+1 = 2(2p^2 + 2p) + 1$ which is an odd number.

Since the statement we have proved (the contrapositive) is logically equivalent to the original statement to be shown, we are done. □

In worksheet 2 you are asked to prove/disprove something similar, $\equiv 0 \mod 3$ instead of $\equiv 0 \mod 2$.

Lemma 4.4. *Let a, b be non-negative real numbers. If $a < b$ then $\sqrt{a} < \sqrt{b}$.*

Is it true? Draw a plot (eg. in Mathematica) to check. To prove, we will show the contrapositive: if $\sqrt{a} \geqslant \sqrt{b}$ then $b \geqslant a$.

[3] Proof by exhaustion (check all possible cases one-by-one), probabilistic (show that the probability of something happening is positive, therefore it must happen), proof by authority, proof by intimidation ....

Proof. First note that if $x \geqslant 0$ then $\sqrt{x}$ is defined and is $\geqslant 0$, so we have $\sqrt{a}, \sqrt{b} \geqslant 0$.

Suppose $\sqrt{a} \geqslant \sqrt{b}$. Then

$$\begin{aligned} a &= \sqrt{a}\sqrt{a} && \text{(by definition of } \sqrt{\cdot}\text{)} \\ &\geqslant \sqrt{a}\sqrt{b} && \text{(by supposition and since } \sqrt{a} \geqslant 0\text{)} \\ &\geqslant \sqrt{b}\sqrt{b} && \text{(by supposition and since } \sqrt{b} \geqslant 0\text{)} \\ &= b && \text{(by definition of } \sqrt{\cdot}\text{).} \end{aligned}$$

The result follows (by contrapositive). □

Putting Lemmas 4.2 and 4.4 together, ie. $(p \to q) \wedge (q \to p)$, we get $p \leftrightarrow q$ or $a < b$ if and only if $\sqrt{a} < \sqrt{b}$.

4.3. **Contradiction.** Recall that the statement $(\neg p \to F) \to p$ is a tautology. (F is a statement whose truth value is always 0 =false).

So, if you want to prove that p is true, you can prove that $\neg p$ implies something that is always false.

Recall that a prime number is an integer $p > 1$ whose only positive divisors are itself and 1. If you start to list them, they seem to appear less and less often the higher you go. So do they run out eventually?

Theorem 4.5 (Euclid)**.** *There are infinitely many different primes.*

This time we have a statement $p =$ "there are infinitely many primes", and we will prove that $\neg p$ implies a contradiction, $\neg p \to F$.

Proof. Suppose (for contradiction) this is not true. So here are all the distinct primes:

$$p_1, p_2, \ldots, p_n.$$

Any other number not on this list is not a prime. Okay, now I will challenge that. Consider

$$N = p_1 p_2 \ldots p_n + 1$$

Well, this is bigger than all the p_i so it is definitely not on the list. So our assumption says it is not prime, so it is equal to some product of smaller numbers. This means, one of the p_i is a divisor. [4] But this is false, because if you divide N by p_i you get a whole number plus $\frac{1}{p_i}$. □

A number x is called *rational* if there exist $a, b \in \mathbb{Z}$ such that $x = \frac{a}{b}$. For example, $0.3333\ldots$ is rational because it is equal to $\frac{1}{3}$. The set of all rational numbers is denoted by $\mathbb{Q}$. A real number that is not in $\mathbb{Q}$ is called *irrational*.

[4] That sentence there needs a mini-proof too: Lemma 5.7.

Lemma 4.6. *$\sqrt{2}$ is irrational.*

See worksheet 2 for a step-by-step guide to proving this.

Many more examples and exercises on writing basic proofs in the homework and worksheets.

5. Induction

An element s in a subset $S \subseteq \mathbb{N}$ is called a *first element* in S if $s \leqslant x$ for every $x \in S$.

Lemma 5.1. *First elements are* unique. *(So we can say "the" first element).*

Proof. Suppose (for contradiction) there was a set $S \subseteq \mathbb{N}$ and two elements $s, t \in S$ both obeying the definition of first element of S. Then since $t \in S$ we have $s \leqslant t$ (thinking of t as "an x" in the definition) and since $s \in S$ we have $t \leqslant s$ (thinking of s as an x). Then $s = t$ so there was only one. □

The following statement is an *axiom* or fact which does not follow from other facts.

Axiom 5.2 (Well ordering principle)**.** Every non-empty subset of $\mathbb{N}$ has a first element.

Eg: $\{5, 4, 6, 7\}$ has a first element, 4.

Lemma 5.3. *Let $n, d \in \mathbb{Z}$ with $d > 0$. Then there exist $q, r \in \mathbb{Z}$ with $0 \leqslant r < d$ such that $n = qd + r$.*

Proof. Define $M = \{n - qd \mid q \in \mathbb{Z}\}$. Then $M \cap \mathbb{N}$ is a subset of $\mathbb{N}$. It is non-empty because if $n \geqslant 0$ you can take $q = 0$ and if $n < 0$ take $q = 2n$ (which is a negative number, so $-qd$ is a big positive number).

Therefore by the well ordering principle $M \cap \mathbb{N}$ has a first element, call it r. Since $r \in M \cap \mathbb{N}$ we have $r \geqslant 0$ and $r = n - qd$ for some $q \in \mathbb{Z}$. If $r \geqslant d$ (for contradiction) then $r - d \geqslant 0$ and $r - d = n - (q + 1)d$ so belongs to $M \cap \mathbb{N}$, and is smaller than r, contradicting our choice of r as first element. □

The well ordering principle is equivalent to another, perhaps more famous, principle:

Axiom 5.4 (Principle of mathematical induction)**.** Let $P(n)$ be a statement about natural numbers $n \geqslant 1$. If

1. $P(1)$ is true
2. $P(k) \to P(k + 1)$ is true

then $P(n)$ is true for all $n \geqslant 1$.

A nice image is an infinite line of dominoes. $P(i)$ is the statement that domino number i falls. Think about how the two conditions show that all dominoes fall.

In the worksheet we will prove that WOP implies PMI. Hint: contradiction and consider the set $\{n \in \mathbb{N}_+ \mid \neg P(n)\}$.

Lemma 5.5. *For all* $n \in \mathbb{N}, n \geqslant 1$

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$$

Proof. Let $P(n)$ be the statement that $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$. Then $P(1)$ is true because LHS is 1 and RHS is $\frac{1(1+1)}{2} = 1$.

Assume that $P(k)$ is true for some $k \geqslant 1$. Then consider $P(k+1)$:

$$\text{LHS} = 1 + 2 + 3 + \cdots + k + (k+1) = \frac{k(k+1)}{2} + (k+1)$$

using the fact that we assume $P(k)$ is true. Manipulate the RHS [5] to get the answer

$$\frac{(k+1)((k+1)+1)}{2}$$

Since $P(1)$ is true and $P(k) \to P(k+1)$ is true for $k \in \mathbb{N}, n \geqslant 1$ then by PMI $P(n)$ is true for all $n \in \mathbb{N}, n \geqslant 1$. □

Exercise 5.6. Prove that for $n \geqslant 1$

$$1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

PMI is equivalent to a seemingly stronger statement: if $P(s)$ is true and if for all $s \leqslant i \leqslant n$ $P(i)$ is true, then $P(n+1)$ is true, then $P(n)$ is true for all $n \in \mathbb{Z}, n \geqslant s$.

Lemma 5.7. *For all* $n \in \mathbb{N}, n > 1$ *if* n *is not prime then some prime number* p *divides* n.

Proof. Let $P(n)$ be the statement that either n is prime or some prime divides n. The statement is true for $n = 2, 3, 4$ ($2, 3$ are primes so it is true, and $n = 4$ is divisible by 2). Assume for all $2 \leqslant i \leqslant n$, if i is not prime then it has a prime divisor. Then if $n+1$ is not prime, by definition $n + 1 = dq$ where $d, q \in \mathbb{N}$ and $1 < d < n + 1$. Since $2 \leqslant d \leqslant n$ then d is either prime or if not prime, some prime p divides it, so $d = pr$, and $n + 1 = dq = prq$ so $p \mid (n+1)$, so $P(n+1)$ is true.

Then by PMI (stronger version) $P(n)$ is true for all $n \geqslant 2$. □

[5] ie. $\frac{k(k+1)}{2} + (k+1) = \frac{k(k+1)}{2} + \frac{2(k+1)}{2} = \frac{k(k+1)+2(k+1)}{2} = \frac{(k+1)(k+2)}{2}$

Lemma 5.8. *For all $n \in \mathbb{N}$, $11^n - 4^n$ is divisible by 7.*

Proof. Let $P(n)$ be the statement that $11^n - 4^n$ is divisible by 7. Then $P(0)$ is true because $11^0 - 4^0 = 1 - 1 = 0$ and $0 = 0.7$ so it is divisible by 7 (by our definition of *divides* from the RSA notes). Let's check $P(1)$ since the 0 case is a bit weird: $11^1 - 4^1 = 7$, great.

Assume that $P(k)$ is true for some $k \geqslant 0$. Then consider $P(k+1)$:

$$11^{k+1} - 4^{k+1} = 11.11^k - 4.4^k$$

what do we do with this? Trick: $11 = 7 + 4$, so write

$$11.11^k - 4.4^k = (7+4).11^k - 4.4^k = 7.11^k + 4.11^k - 4.4^k = 7.11^k + 4(11^k - 4^k)$$

now by our assumption $(11^k - 4^k) = 7s$ for some s, so

$$7.11^k + 4(11^k - 4^k) = 7.11^k + 4.7s = 7(11^k + 4s)$$

so is divisible by 7.

Since $P(0)$ and $P(k) \to P(k+1)$ are both true, by PMI $P(n)$ is true for all $n \in \mathbb{N}$. □

Lemma 5.9. *All horses are black.*

Proof. Let $P(n)$ be the statement that in any set of n horses, all the horses in that set are black. $P(0)$: if you take no horses, then every horse contained in that set is black. (Since there are no horses, this is true. The contrapositive of *for all horses in my set, the horse is black* is *there exists a horse in my set which is not black.*)

$P(k+1)$: take a set of $k+1$ horses, remove one, then this is a set of k horses, which by assumption is all black. Now maybe the horse we removed was white. So put it back in, and remove a different horse, again you get a set of k horses so they must all be black. □

6

6. Correctness of computer programs

How do you *prove* some code really does what you say it does? How do you prove it will terminate?

We use what is called a *loop invariant*, some fact that is true at the start of the code, and if it is true before one iteration of a loop then it is true after. If you can also show some quantity is getting smaller, or some other way to show the loop will stop after a finite number of steps, this together with the loop invariant (which will still hold at the end by induction) show correctness.

[6] in my proof, I needed two distinct horses to be in my set. If I only had one (white) horse, I remove it to get an empty set, which is all black ($P(0)$), but when I put it back in, I don't have any other horse to remove.

Example 6.1. Consider the following somewhat useless fragment of code

```
int j = 9;
for(int i=0; i<10; i++)
   j--;
```

We can define a loop invariant for it to be the statement that $i+j = 9$. This is true before the loop (when $i = 0$ and $j = 9$), and if $i + j = 9$ then applying `i++` followed by `j--` means the new values of i and j still sum to 9.

In Lemma 5.3 we proved that for every integer x and positive integer d you can find q, r with $0 \leqslant r < d$ and $x = qd + r$. In fact more is true: the q and the r that we get are *unique*.

Proof. Suppose there are two sets of numbers that work, so $x = q_1 d + r_1 = q_2 d + r_2$.

Then $(q_1 - q_2)d = r_2 - r_1$. But since $0 \leqslant r_1, r_2 < d$ the right side of this equation is strictly between $-d$ and d. The left side is an integer multiple of d, so the only number strictly between $-d$ and d it can be is 0.

Thus $r_2 - r_1 = 0$ so $r_2 = r_1$, and $(q_1 - q_2) = 0$ so $q_1 = q_2$. □

This proof and the proof of Lemma 5.3 tell us the unique numbers q, r *exist*, but how do we actually find them? That is, the well ordering principle just says sets have first elements, it doesn't tell you what the first element is. Luckily it is easy to compute q, r: assume x, d are positive (for simplicity) then run the following

```
q=0;
r=x;
while(r>=d)
  r=r-d;
  q++;
return (q,r)
```

The while loop stops because r is getting smaller each iteration. The loop invariant is ($x = qd+r$ and $r \geqslant 0$): this is true at the start because $q = 0$ and $r = x$. In one iteration we subtract d from r but q increases by 1 so qd increases by d, so the sum stays the same. Since the loop is only entered if $r \geqslant d$ the r stays non-negative in one iteration of the loop.

Example 6.2. Recall the Euclidean algorithm which (allegedly) computes $\gcd(a, b)$. (See Section 14)

We can now prove it is correct. Again I will assume $a, b \geqslant 0$ for simplicity. First, we have an algorithm which computes the remainder $0 \leqslant r < y$ of x on division by y (assuming $x \geqslant y \geqslant 0$).

Let $a, b \in \mathbb{Z}$ and $a \neq 0 \vee b \neq 0$.

The steps are:

(1) Start with (a, b) such that $a \geqslant b$. (ie. put them in order).
(2) While $b \neq 0$,
    compute the remainder $0 \leqslant r < b$ of a divided by b.
    set $a \leftarrow b, b \leftarrow r$ (and thus $a \geqslant b$ again).
(3) Return a

First, if $b = 0$ at the start then we skip the while loop and return a, which is the $\gcd(a, 0)$.

We claim that the loop invariant is the value $\gcd(a, b)$. To prove this claim, (its clearly true at the start) we need to show if $a = qb + r$ with $0 \leqslant r < b$ then $\gcd(a, b) = \gcd(b, r)$. Suppose this is not true, let $d = \gcd(a, b)$ and $c = \gcd(b, r)$. Then d divides r and c divides a, so they are both common divisors of all three numbers a, b, r. If d divides c, then as $d = \gcd(a, b)$ and c divides both a and b then $d = c$ by definition of gcd. Similarly $(p \vee \neg p)$ if c divides d we get the same result. So $d = c$.

Part 2. Sets, functions, complexity

Part 2 covers the basics of set theory, functions between sets, applications to counting and algorithm complexity. Once we have the right fundamentals we can easily talk formally and prove things about about graphs, trees, (block designs), etc.

We have been using notation like $A \subseteq B, x \in A, x \in \mathbb{N}, A \cap B, |A|$ without properly defining it. Here we start by defining sets, seeing the connection with logic we studied before, then functions between sets, which will enable us to talk correctly and clearly about graphs and trees, algorithm complexity analysis, block designs, and beyond the course whenever you encounter any mathematical statements.

7. Set theory

A *set* is a well-defined collection of objects. The objects are called *elements* of the set, or *members* of the set. Carefully defining what *well-defined* means will take us beyond the scope of this course, into axiomatic set theory and foundations of mathematics.

We can represent a set using brackets, for example $A = \{1, 2, a, 5, c\}$. The elements are the five symbols you see listed inside the brackets.

We could also describe a set using variables satisfying some conditions, for example:

$$B = \{x \mid (x \in \mathbb{N}[(1 \leqslant x \leqslant 5) \wedge (x \neq 4)]) \vee (x = a) \vee (x = c)\}.$$

The set $\{1, 5, 3, c, a, 1, 2\}$ is the same as the set B, since a set is defined only by the elements it contains, no matter how they are listed or displayed.

The notation $x \in A$ means x *is an element of* A and $x \notin A$ means $\neg(x \in A)$.

Example 7.1. (1) $C = \{x \mid x \in \mathbb{R}, x < 0\}$
(2) $D = \{y \mid y \in \mathbb{R}, y^2 = 2\}$
(3) $A \cap B = \{x \mid x \in A \wedge x \in B\}$ (intersection)
(4) $A \cup B = \{x \mid x \in A \vee x \in B\}$ (union)

Note the similarity of notation for $\cap$ and $\wedge$, and $\cup$ and $\vee$. In the example above $C \cap D = \{-\sqrt{2}\}$ since this is the only number that is in both sets.

Exercise 7.2. Let $A = \{a, b, c, d, e\}, B = \{b, d, e\}, C = \{f, g, a\}$. Find
(1) $(A \cup B) \cap (A \cup C)$
(2) $A \cap (B \cup C)$

A pictorial way to do this exercise is to draw a *Venn diagram*. See the worksheet.

If A, B are sets we say A is a *subset* of B if $\forall x \in A, x \in B$, or $(x \in A) \to (x \in B)$. Notation $A \subseteq B$. The notation $A \subset B$ means *strictly contains*:

$$((x \in A) \to (x \in B)) \wedge (\exists y[y \in B \wedge y \notin A]).$$

So $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$.

Let $\mathscr{U}$ be some large "universal" set, so we assume all sets we speak about are subsets of $\mathscr{U}$. Then $\overline{A} = \{x \mid x \notin A\}$ means the set of elements in $\mathscr{U}$ that are not in A.

There is a strong connection to the logic we covered before. We have three operations on sets: $\cap, \cup, ^-$ which we can use to build new sets from old ones, and in logic we have three connectives $\wedge, \vee, \neg$.

Recall the tautologies in logic such as

$$\neg(p \wedge q) \leftrightarrow \neg p \vee \neg q$$

In set theory we could consider sets

$$\overline{A \cap B} \text{ and } \overline{A} \cup \overline{B}.$$

How do we show two sets are the same? We show they contain exactly the same elements.

Lemma 7.3. $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

The proof goes: pick some arbitrary element of the LHS. Show it belongs to the RHS. Since we picked an arbitrary thing, this shows everything in the LHS is also in the RHS, so LHS⊆RHS. Repeat to get RHS⊆LHS, then LHS=RHS.

Proof. Suppose $x \in \overline{A \cap B}$. Then x is not in $A \cap B$. Now either $x \in A$ or not. If $x \in A$ then since $x \notin A \cap B$ we must have x is not in B. So either $x \in \overline{A}$ or $x \in \overline{B}$, so $x \in \overline{A} \cup \overline{B}$. Thus

$$\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}.$$

Next, start over and suppose $x \in \overline{A} \cup \overline{B}$. If $x \in A$ then $x \in \overline{B}$ so $x \notin A \cap B$. Otherwise $x \notin A$ so $x \notin A \cap B$. In both cases we have $x \notin A \cap B$ so $x \in \overline{A \cap B}$. Thus

$$\overline{A} \cup \overline{B} \subseteq \overline{A \cap B}.$$

Since each set is contained in the other, they are equal. □

Exercise 7.4. Show that for any sets $A, B, C \subseteq \mathscr{U}$

$$A \cap (B \cup C) = (A \cup B) \cap (A \cup C).$$

See the worksheet for a list of set identities next to a list of logical tautologies.

The next exercise explains why *well-defined collection of objects* is not quite good enough.

Exercise 7.5. Let $P(S)$ be the property (of sets) that S does not contain itself. For example, $P(\mathbb{N})$ is true because $\mathbb{N}$ contains numbers, it does not contain sets so it cannot contain itself. Another example: the *empty set* $\emptyset$ is the set that has no elements, $\emptyset = \{\}$. So it contains nothing so cannot contain itself.

(a) Give some more examples.

Consider the set of all abstract concepts. Call it $\mathscr{A}$. Then A contains things like art, postmodernism, democracy, imaginary numbers.

(b) Which is true: $\mathscr{A} \in \mathscr{A}$ or $\mathscr{A} \notin \mathscr{A}$?

Let $\mathscr{S} = \{S \mid P(S)\}$ be the set of all sets that do not contain themselves.

(c) Which is true: $\mathscr{S} \in \mathscr{S}$ or $\mathscr{S} \notin \mathscr{S}$?

7.1. **Power set.** Let A be a set. Then (axiom)[7]

$$\mathscr{P}(A) = \{B \mid B \subseteq A\}$$

is a set. Its called the *power set* of A.

Questions:

- is $\emptyset \in \mathscr{P}(A)$?
- is $A \in \mathscr{P}(A)$?
- is $\mathscr{P}(A) \in \mathscr{P}(A)$?

[8]

Exercise 7.6.
- Given $A = \{1, 2, 3\}$ is a set, what is $\mathscr{P}(A)$?
- Prove that if A is a set then $A \subsetneq \mathscr{P}(A)$
- If A contains 4 elements, how many elements in $\mathscr{P}(A)$?

Another axiom: $\emptyset$ is a set.

What can you build with just these two axioms? We have $\emptyset$ is different to $\mathscr{P}(\emptyset) = \{\emptyset\}$. So we have at least two different sets in our world[9]. Can you think of how to get a third set which is different from these two?

More info here: [4] (advanced topic, just in case you are interested in where this basic set theory can end up). For a 3rd year level textbook on axiomatic set theory see [1].

8. Relations

If A, B are sets we can define a new symbol (a, b) where $a \in A$ and $b \in B$. This symbol is not the same as $\{a, b\}$, it is a new symbol. Also it is not the same as (b, a), the symbol has an *order*. We call it an *ordered pair*. Define $A \times B = \{(a, b) \mid a \in A, b \in B\}$.

Example 8.1. If $A = \{1, 2, 3\}$ and $B = \{d, e\}$ then

$$A \times B = \{(1, d), (2, d), (3, d), (1, e), (2, e), (3, e)\}.$$

A subset of $A \times B$ is called a *relation* from A to B. We often use the notation $\mathscr{R}$ to denote a relation.

---

[7] When we say "axiom", we mean that the following fact is declared to be true. As we have just seen, declaring that for any condition P the collection of objects $\{x \mid P(x)\}$ is a set leads to a contradiction, so we do not wish to make this an axiom.

[8] Answers: Yes, yes, no.

[9] Our world, for now, is only the sets that we can really know exist assuming only our two axioms.

Example 8.2. Let $A = \{1, 2, 3, 4\}$ and define $\mathscr{R} \subseteq A \times A$ by $\mathscr{R} = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$. We write $a\mathscr{R}b$ if $(a, b) \in \mathscr{R}$. So for example $1\mathscr{R}3$.

In fact, the relation in this example could be called *being strictly smaller than*. In general, relations don't have to have any meaning, they are just subsets of $A \times B$.

Exercise 8.3. Let $A = \{1, 2, 3, 4\}$. Write down a relation $\mathscr{R} \subseteq A \times A$ which is like "$\geqslant$".

Definition 8.4. Let A be a set. Then $\mathscr{R} \subseteq A \times A$ is

- *reflexive* if for all $a \in A$, a
- *symmetric* if for all $a, b \in A$, a implies b
- *antisymmetric* if for all $a, b \in A$, a and b implies $a = b$
- *transitive* if for all $a, b, c \in A$, a and b implies a

Exercise 8.5. Let $A = \{1, 2, 3, 4\}$ and

$$\mathscr{R} = \{(1, 1), (2, 2), (3, 1), (1, 3), (2, 3), (3, 2)\}.$$

Decide which of the four properties (reflexive, symmetric, antisymmetric, transitive) $\mathscr{R}$ satisfies.

Exercise 8.6. Construct an example (that means tell me a set A and some subset of $A \times A$) of a relation which is

- *both* symmetric and antisymmetric
- *neither* symmetric nor antisymmetric

These notions are extremely useful throughout mathematics. For now, you should feel good if you can read the very abstract definitions (written in logic and set theory notation) and write down examples, prove/disprove some relation has them. This shows you are "getting it" in this course.

Definition 8.7. Let A be a set. Then $\mathscr{R} \subseteq A \times A$ is

- an *equivalence relation* if it is reflexive, symmetric and transitive
- a *partial order* if it is reflexive, antisymmetric and transitive

Exercise 8.8. Show that "$\equiv \mod d$" is an equivalence relation on $\mathbb{Z}$, and that "$\leqslant$" is a partial order (on $\mathbb{Z}, \mathbb{R}$ etc).

Add: partitions.

9. Functions on sets

A *function* from A to B is a relation $f \subseteq A \times B$ in which every element of A appears exactly once as the first component of an ordered pair in the relation.

That is,

$$\forall a \in A \exists b \in B[(a,b) \in f] \wedge [((a_1,b) \in f \wedge (a_2,b) \in f) \to (a_1 = a_2)]$$

We could also write the condition as

$$\forall a \in A \exists! b \in B[(a,b) \in f]$$

where the notation ! means "unique". So $\exists!$ means "there is exactly one".

Since each $a \in A$ goes to exactly one $b \in B$ we can also use the notation $f(a) = b$, and we write $f : A \to B$.

Example 9.1. Define $f : \mathbb{R} \to \mathbb{Z}$ by

$$f(x) = \lfloor x \rfloor = \text{ the greatest integer less than or equal to } x.$$

Similarly we have $g : \mathbb{R} \to \mathbb{N}$ by

$$g(x) = \lceil x \rceil = \text{ the smallest integer greater than or equal to } x.$$

Exercise 9.2. Let $h : \mathbb{N} \to \mathbb{N}$ defined by

$$h(n) = \left\lceil \frac{n}{2} \right\rceil + 7.$$

[10] If $n =$ your age, compute $h(n)$.

Example 9.3. Let $S =$ the set of all students at UTS and $f \subseteq S \times \mathbb{N}$ where (s,n) means n is a student ID number for student s. What if f was not a function? What if $(s, 13645)$ and $(t, 13645)$ were both in f?

Definition 9.4. Let $f : A \to B$ be a function from a set A to a set B. We say f is *one-to-one* (or 1-1) if

$$\forall x \forall y \in A[f(x) = f(y) \to x = y].$$

We want the student number function to be one-to-one.

Definition 9.5. Let $f : A \to B$ be a function from a set A to a set B. We say f is *onto* if

$$\forall b \in B \exists a \in A[f(a) = b].$$

[10] `https://en.wikipedia.org/wiki/Age_disparity_in_sexual_relationships`. Should it be $\lceil \cdot \rceil$ or $\lfloor \cdot \rfloor$?

Setting up our definitions using logical statements like this, it is easy to prove examples satisfy them or not. For example, $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = x^2$ is not 1-1:

$$\begin{aligned} &\neg\forall x\forall y \in A[f(x) = f(y) \to x = y] \\ &\leftrightarrow \exists x\exists y \in A[f(x) = f(y) \wedge x \neq y] \end{aligned}$$

this is true for $x = 1, y = -1$.

and not onto:

$$\begin{aligned} &\neg\forall b \in B\exists a \in A[f(a) = b] \\ &\leftrightarrow \exists b \in B\forall a \in A[f(a) \neq b] \end{aligned}$$

is true because there exists $b = -1$ so that no real number squared is equal to -1.

Example 9.6. Is the function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = 5x + 3$ one-to-one?

$$\begin{array}{rrcl} & f(x_1) & = & f(x_2) \\ \to & 5x_1 + 3 & = & 5x_2 + 3 \\ \to & 5x_1 & = & 5x_2 \\ \to & x_1 & = & x_2 \end{array}$$

Since x_1, x_2 are arbitrary, the condition holds for all $x_1, x_2 \in \mathbb{R}$ so f is 1-1.

Exercise 9.7. Is the function h from Exercise 9.2 one-to-one? (Prove or disprove).

Exercise 9.8. Let $A = \{a, b, c, d, e\}, B = \{b, d, e\}, C = \{f, g, a\}$. Give examples of functions

(1) $f : A \to B$ which is onto and not 1-1
(2) $g : A \to B$ which is 1-1 and not onto
(3) $h : A \to B$ which is both 1-1 and onto
(4) $i : B \to C$ which is onto and not 1-1
(5) $j : B \to C$ which is 1-1 and not onto
(6) $k : B \to C$ which is both 1-1 and onto

Lemma 9.9. *Let A, B be finite sets. If $f : A \to B$ is*

- *1-1 then $|A| \leqslant |B|$.*
- *onto then $|B| \leqslant |A|$.*

Proof. For 1-1 see pigeonhole principle section. The proof goes: if $|A| > |B|$, then we have $|A|$ pigeons to be placed into $|B|$ holes, but since we have more pigeons than holes, there must be at least one hole with more than one pigeon in it. So if we place pigeon a in hole $b = f(a)$ there will be some b with $b = f(a_1) = f(a_2)$.

For the second claim, if $|B| > |A|$ then write

$$B = \{f(a) \mid a \in A\} \cup \{b \mid \nexists a \in A[f(a) = b]\}.$$

The size of the first set is at most A, so the second set contains at least one element, so f is not onto. So we get the second claim by contrapositive. □

Definition 9.10. A function $f : A \to B$ is a *bijection* if it is both 1-1 and onto.

Exercise 9.11. Show that $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = 5x + 3$ is a bijection.

Example 9.12. $\mathbb{N}$ and $\mathbb{Z}$ are in bijection. What? What about Lemma 9.9? Surely $\mathbb{N}$ is smaller than $\mathbb{Z}$?

Here is a bijective function.

$$f(n) = \begin{cases} k & n = 2k \\ -k & n = 2k+1 \end{cases}$$

Check it. (check it really is a function from $\mathbb{N}$, it is 1-1 and onto.)

What does this mean? It means we haven't defined *size* of infinite sets properly yet. In fact, we will define two sets to be the same size if there is a bijection between them.

Exercise 9.13. Is there a bijection between $\mathbb{N}$ and $\mathbb{Q}_+$?

Example 9.14. Is there a bijection between $\mathbb{N}$ and $\mathbb{R}_+$?

See the lecture slides for hints to these.

If $f : A \to B$ is a function between sets, and $A_1 \subseteq A$, define the notation $f(A_1) = \{b \in B \mid \exists a \in A_1[f(a) = b]\}$. Then using this notation, f is onto if and only if $f(A) = B$.

Exercise 9.15. Let $f : A \to B$ be one-to-one and $A_1, A_2 \subseteq A$. Prove that $f(A_1 \cap A_2) = f(A_1) \cap f(A_2)$. Is the statement still true if f is not one-to-one?

Exercise 9.16. Define a function $A : \mathbb{N}^2 \to \mathbb{N}$ using the following *recursive* definition.

$$\begin{array}{lcll} A(0,n) & = & n+1 & n \geqslant 0, \\ A(m,0) & = & A(m-1,1) & m > 0, \\ A(m,n) & = & A(m-1, A(m,n-1)) & m,n > 0. \end{array}$$

(a) Compute $A(1,3)$.
(b) Compute $A(2,3)$.
(c) Prove that $A(1,n) = n+2$ for all $n \in \mathbb{N}$.

(d) Prove that $A(2, n) = 3 + 2n$ for all $n \in \mathbb{N}$.
(e) Prove that $A(3, n) = 2^{n+3} - 3$ for all $n \in \mathbb{N}$.

This function is called *Ackermann's function*, it is famous because beyond these examples it is really hard to compute (it grows really fast with its inputs). There are many different versions.

10. Algorithm complexity

This is a very brief introduction, if you study computer science you will see this in more detail later. We first need a notion of comparing speed (or memory usage) of algorithms. Imagine running an algorithm on an old Windows 98 computer, and running the same algorithm on my new MacBook. The algorithm is the same, but the run times will be different. We want a notion of comparing algorithms that ignores particular computer implementations, and just says "doing it this way is faster than doing it this way".

Example 10.1. Consider these two algorithms in Murray-pseudocode:

```
power_slow(a real; n positive integer)
x = 1.0
for i = 1 to n do
    x = x*a
return x
```

```
power_fast(a real; n positive integer)
x = 1.0
i = n
while i > 0 do
    if i is odd then
        x = x*a
    i = floor(i/2)
    if i > 0 then
        a = a*a
return x
```

If we compute `power_slow`$(3, 6)$ the computation takes 6 steps:

`i`	`x`
	1
1	3
2	9
3	27
4	81
5	243
6	729

If we compute `power_fast`$(3, 6)$ the computation takes 3 steps:

`x`	`i`	`a`
1	6	3
1	3	9
9	1	81
729	0	

We might argue about whether this is 3 or 4 steps, but ignoring actual implementation details and precise ways to count, we can see that the fast algorithm will be qualitatively faster for large n. The idea of the next definition is to just capture the speed without worrying too much about details. The speed depends on the input n (and the input value a but let's ignore that). We also don't want to worry about small inputs, where maybe the slow algorithm happens to be better. We want to say "doing it this way, in general, for large n, is *this* fast".

Definition 10.2. Let $f, g : \mathbb{N}_+ \to \mathbb{R}$. We say that g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{Z}^+$ such that

$$|f(n)| \leqslant m|g(n)|$$

for all $n \in \mathbb{N}, n \geqslant k$. We use the notation $f \in O(g)$ and read this as "f is in Big O of g".

Example 10.3. The function $f(n) = \log_2 n + 5$ is dominated by $g(n) = n$, since for $n \geqslant 10$ we have $\log_2(n) + 5 \leqslant n$. In fact we could choose k smaller, but 10 is good enough: $\log_2(10) \approx 3.32$.[11]

Some careful analysis of the two algorithms above shows that, up to some constants, the fast version takes about $\log_2 n$ steps (because each step divides i by 2, roughly).

[11]Wait, my calculator doesn't have a button for $\log_2$ only ln or $\log_{10}$?! See Example 10.4 next.

Example 10.4. Recall the formula

$$\log_b y = \frac{\log_a y}{\log_a b}.$$

If $b = 2$ and your calculator only has a button for $\log_{10}$ or $\log_e = \ln$ then you can use this formula:

$$\log_2 n = \frac{\log_e n}{\log_e 2}.$$

Since $\log_e 2 \approx 4.6$ is a fixed number, it doesn't matter which log we use because of the m in the definition.

Exercise 10.5. Let $f(n) = 6n$ and $g(n) = n^2$. Show that g dominates f, that is, $6n \in O(n^2)$.

In logic notation $f \in O(g)$ if

$$\exists m \in \mathbb{R}_+ \exists k \in \mathbb{N}_+ \forall n \in \mathbb{N}_+ [(n \geqslant k) \to (|f(n)| \leqslant m|g(n)|)].$$

The negation $f \notin O(g)$ is

$$\forall m \in \mathbb{R}_+ \forall k \in \mathbb{N}_+ \exists n \in \mathbb{N}_+ [(n \geqslant k) \wedge (|f(n)| > m|g(n)|)].$$

This means, given any constants m, k we can find some n which messes it up: n is bigger than k and $f(n)$ is bigger than $mg(n)$.

Example 10.6. Let $f(n) = 6n$ and $g(n) = n^2$. Show that g is not dominated by f. Using logic notation we want to show

$$\forall m \in \mathbb{R}_+ \forall k \in \mathbb{N}_+ \exists n \in \mathbb{N}_+ [(n \geqslant k) \wedge (|g(n)| > m|f(n)|)].$$

Given m, k, there exists (we can choose) $n > \max\{6m, k\}$. Then $n \geqslant k$ and $|g(n)| = n^2 > 6mn$ since $n > 6n$ so

$$|g(n)| = n^2 > 6mn = m|6n| = m|f(n)|.$$

Alternatively we can argue by contradiction: suppose $g \in O(f)$, then there is some m, k so that $n^2 \leqslant 6mn$ for all $n \geqslant k$. Then since n is positive, dividing both sides by n we get $n \leqslant 6m$, which contradicts that is holds for all n larger than k.

In the workshop, you can get some practice showing which functions are "bigger" than which, and which are the same. Here are some warm-ups to think about:

Exercise 10.7. Let $f(n) = 6n^2 + 5n + 2$ and $g(n) = n^2$. Show that $f \in O(g)$ and $g \in O(f)$. So they are (up to Big O equivalence) the "same".

Exercise 10.8. Show that $f(n) = n^c$ is dominated e^n where c is any positive integer. So polynomials are "slower" than exponentials.

Exercise 10.9. Which is faster (who dominates who?) – exponentials or factorials? [12]

Big-O form	Name
$O(1)$	constant
$O(\log_2 n)$	logarithmic
$O(n)$	linear
$O(n \log_2 n)$	$n \log n$ (sometimes called quasilinear)
$O(n^2)$	quadratic
$O(n^3)$	cubic
$O(n^p), p \in \mathbb{N}$	polynomial
$O(c^n), c > 1$	exponential
$O(n!)$	factorial

TABLE 1. Some standard functions for comparison.

11. The pigeonhole principle

Before we needed to prove that if A, B were finite sets, and A was smaller than B, then no function from A to B could be 1-1. If you tried to make a function, some $a \in A$ would have to be sent to some $b \in B$ that was already taken. To prove this intuitive hunch, we need an axiom:

Axiom 11.1 (Pigeonhole principle)**.** If m pigeons occupy n pigeonholes and $m > n$ then some pigeonhole has at least two pigeons in it.

This says, no matter how you try to arrange, maybe some holes are empty, but no matter always one will have two or more pigeons.

Exercise 11.2. In your workshop, how many people are present? What is the chance two people were born in the same month?

Axiom 11.3 (Pigeonhole principle)**.** If m pigeons occupy n pigeonholes and $m > kn$ then some pigeonhole has more than k pigeons in it.

Example 11.4. Let $A \subseteq \mathbb{N}_+$ with $|A| = 28$. Then A contains at least two elements with the same remainder mod 27. Proof: the pigeons are the elements of A. The pigeonholes are the 27 possible values of remainder when you write $x = 27q + r$. Since we have 28 numbers to fit into 27 holes, the result follows by PHP.

[12] that is, show that one of $f(n) = e^n$, $g(n) = n!$ dominates the other.

Exercise 11.5. Frank is in a rush and grabs socks one at a time at random from the dryer. He knows he washed 10 distinct pairs of socks the day before and put them in the dryer. How many socks will he need to take out before he is guaranteed to have a matching pair?

Exercise 11.6. Let p be an odd number and let S be any subset of $\mathbb{Z}_{p+1} = \{0, 1, \ldots, p\}$ which contains at least

$$2 + \frac{p-1}{2}$$

elements. Show that there are at least two elements of S with sum equal to p.

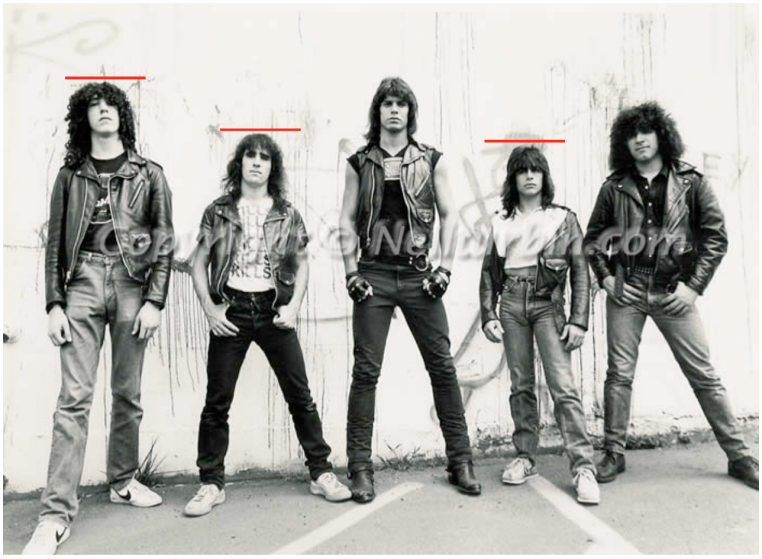
Recall Lemma 9.9. We can now understand the proof given before.

Lemma 11.7. *Let A, B be finite sets. If $f : A \to B$ is 1-1 then $|A| \leqslant |B|$.*

Proof. If $|A| > |B|$, then we have $|A|$ pigeons to be placed into $|B|$ holes, but since we have more pigeons than holes, there must be at least one hole with more than one pigeon in it. So if we place pigeon a in hole $b = f(a)$ there will be some b with $b = f(a_1) = f(a_2)$ and thus f is not 1-1. So by contrapositive the statement of the lemma is true. □

We say a sequence $3, 9, 2, 7, 6, 1, 4, 10, 5, 8$ *contains a subsequence* if you can remove a few numbers to obtain the subsequence, for example the above contains $2, 6, 4, 8$. It contains an increasing sequence $3, 6, 10$ and a decreasing sequence $9, 7, 6, 4$. Claim:

Exercise 11.8. For each $n \in \mathbb{N}_+$, any sequence of $n^2 + 1$ distinct real numbers contains a decreasing or increasing subsequence of length $n + 1$.



Step by step hints are given in Worksheet 6 (bonus question, see online). Before you look at the hints, think about how you might prove (or disprove) this.

Part 3. Counting

inclusion-exclusion
permutations, combinations. Number of license plates etc
Fibonacci (how many strings don't contain 11, start with 1, etc)
Catalan - lattice paths, binary trees, brackets

Part 4. Introduction to number theory

Notes to learn enough number theory to be able to properly understand RSA cryptography.

12. Modular arithmetic: ISBN and credit card check digits

Friendly intro using ISBN and CC. Talk about QR codes.
(to be added)

13. gcd

For this entire section, all numbers d, a, b, n etc. are *integers*.

Definition 13.1. Let $d, a, b \in \mathbb{Z}$. We call d a *divisor* of a if $a = ds$ for some $s \in \mathbb{Z}$. We write $d \mid a$. We call d a *greatest common divisor* of a and b if

- $d \mid a$ and $d \mid b$ (common divisor)
- if $c \mid a$ and $c \mid b$ then $c \mid d$ (greatest)

We write $d = \gcd(a, b)$.

For example, $\gcd(21, 39)$: 1 divides both (always true) but so does 3, and $1 \mid 3$ so $\gcd(21, 39) = 3$.

14. Euclidean algorithm

To find the gcd we have this (very old) algorithm:
Write

$$39 = \_\_\_ \times 21 + \_\_\_$$

where the last gap is a *remainder* between 0 and 29.

$$39 = 1 \times 21 + 18$$

Then repeat

$$21 = \_\_\_ \times 18 + \_\_\_$$

Repeat until the remainder is 0.
Why does this algorithm work?

$$39 = 1 \times 21 + 18$$
$$21 = 1 \times 18 + 3$$
$$18 = 6 \times 3 + 0$$

The last line says 3 divides 18 (because the remainder is 0) and the second last line has 18 and 3 on the right hand side, and since 3 divides both of these terms, 3 divides the left side.

Repeat this argument until the first line, so 3 divides both 39 and 21.

Why does the algorithm find the *greatest* divisor? See algorithm complexity section.

Example: find $\gcd(26, 81)$

$$81 = 3 \times 26 + \_\_\_$$

(always start with the big number on the left) repeat until remainder is 0

$$26 = \_\_\_ \times \_\_\_ + \_\_\_$$
$$\_\_\_ = \_\_\_ \times \_\_\_ + \_\_\_$$

$$\vdots$$

$$\_\_\_ = \_\_\_ \times \_\_\_ + 0$$

15. Euclidean algorithm backwards

Using the steps of the algorithm, you can always write $\gcd(a, b) = \lambda a + \mu b$ for some numbers λ, μ.

Example

$$81 = 3 \times 26 + 3$$

$$26 = 8 \times 3 + 2$$

$$3 = 1 \times 2 + 1$$

$$2 = 2 \times 1 + 0$$

so $\gcd(81, 26) = 1$, now use the steps backwards:

$$1 = 3 - 2 = 3 - (26 - 8.3) = (1 + 8)3 - 26$$

$$= 9.3 - 26$$

$$= 9.(81 - 3.26) - 26$$

$$= 9.81 - 28.26$$

Exercise: find μ and λ integers so that $3 = \mu(21) + \lambda(18)$.

Definition 15.1. If $\gcd(a, b) = 1$ we say that a and b are *relatively prime*.

16. Euler's ϕ function

Definition 16.1. Let $n \in \mathbb{Z}$. Define $\phi(n)$ to be the number of positive integers less than n that are relatively prime to n.

In mathematical notation,

$$\phi(n) = |\{i \mid 1 \leqslant i \leqslant n, \gcd(i, n) = 1\}|$$

Example: $\phi(9) = |\{1, 2, 4, 5, 7, 8\}| = 6$. Find $\phi(7)$.

If a number p is *prime* the every number from 1 to $p - 1$ is relatively prime to it, so $\phi(p) = p - 1$.

Exercise: show that $\phi(p^2) = p(p-1)$. Hint: write out all the numbers in a nice table form, and see that everything except $p, 2p, 3p, \ldots, (p-1)p$ is relatively prime.

Fact (to prove): If a and b are relatively prime then $\phi(ab) = \phi(a)\phi(b)$.

17. Repeated squaring

Recall worksheet 1, and assessment task 1, we had an efficient way to compute things like

$$121^{12} \mod 13.$$

First, we had

$$[xy]_n = [[x]_n[y]_n]_n$$

(the remainders $\mod n$ are the same if you multiply first or second).

Then

$$121 \equiv 4 \mod 13$$

so $121^{12} \equiv 4^{12} \mod 13$. Now we do *repeated squaring*

$$4^2 = 16 \equiv 3 \mod 13, 4^4 = 4^2.4^2 \equiv 3.3 = 9,$$

$$4^8 = 4^4.4^4 \equiv 9.9 = 81 \equiv 3 \mod 13$$

so

$$4^{12} = 4^8.4^4 \equiv 3.9 = 27 \equiv 1$$

General fact (interesting to prove): If a, n are relatively prime then

$$a^{\phi(n)} \equiv 1 \mod n$$

.

Let's prove it in the special case when n is a prime, call it p. (In this case any $1 \leqslant a < p$ is relatively prime to p, and $\phi(p) = p - 1$).

Theorem 17.1 (Fermat's little theorem)**.** *If p is prime and $0 < a < p$ then*

$$a^{p-1} \equiv 1 \mod p.$$

Proof. Write

$$a, 2a, 3a, \ldots, (p-1)a.$$

(Why? You will see, its a clever trick).

Now, we claim that all of these numbers are distinct when you reduce mod p. To show this, suppose $ra \equiv sa \mod p$ for some r, s between 1 and $p - 1$. Then $ra = sa + kp, (r - s)a = kp$ so p divides both sides, so p either divides a (which it can't) or it divides $r - s$. However $r - s$ is a number strictly between $-p$ and p, so the only possibility is that $r - s = 0$ so $r = s$.

This shows that all those numbers are distinct $\mod p$.

Example: $p = 7$ and $a = 2$, we get $2, 4, 6, 8, 10, 12$ which reduces to $2, 4, 6, 1, 3, 5$. We don't normally put examples inside proofs.

Now, multiply them all together. Since there are $p-1$ of them, and they are all between 1 and $p-1$ (positive remainders after dividing by p) then

$$a(2a)(3a)\dots((p-1)a) \equiv 1.2.\dots.(p-1)$$

where the numbers on the right side have been moved into this nice order (they would be all mixed up as in the example).

Then taking all the as to the front of the left side we get

$$a^{p-1}(1.2.\dots.(p-1)) \equiv 1.2.\dots.(p-1) \mod p$$

and so the result $a^{p-1} \equiv 1 \mod p$ follows. □

18. RSA

All of the above is very abstract, useless mathematics right?

RSA uses it all to enable secret messages to be sent over public channels, so anyone can see the encrypted messages.

Here is how it works:

(1) Alice picks $n = pq$ where p, q are large distinct primes.
(2) Alice picks d, e such that $de \equiv 1 \mod (p-1)(q-1)$. How? She picks e relatively prime to $(p-1)(q-1)$, and uses the Euclidean algorithm as we did above to write $1 = \lambda e + \mu(p-1(q-1)$ and so the number $d = \lambda$.
(3) Alice publishes n and e, say online. Everyone in the world knows what they are, but they do not know p, q or d. Alice then gives these instructions, again online: "To send me a secret message m $(1 < m < n)$, send me $[m^e]_n$".
(4) Bob wants to send a message m, so he computes $[m^e]_n = c$ and sends it. Anyone can see this remainder, so number between 1 and n, but this remainder is not the same as m, so they don't know what m was.
(5) Alice has more information that everyone else: she knows d. She computes $[c^d]_n$ and by all the theory above, she knows the answer is actually Bob's message m.

How does the last step work?

$$c^d \equiv (m^e)^d = m^{ed} = m^{1+k(p-1)(q-1)}$$

since $ed \equiv 1 \mod (p-1)(q-1)$.

Now what do you notice? Hint: $(p-1)(q-1) = \phi(pq)$.

19. example of RSA

Alice picks $p = 3, q = 5$ so $n = 15$. She needs to pick d, e with $de \equiv (3-1)(5-1) = 2.4 = 8$. She wants to use Euclidean algorithm on $e = 15$ to find d, so she needs some e that is relatively prime to 8:

$$e = 1, 3, 5, 7.$$

If she picks $e = 3$ then $d = 3$ (Euclidean alg). So Alice publishes $e = 3, n = 15$.

Now Bob wants to send $m = 7$. He computes $7^3 = 49.7 \equiv 4.7 = 28 \equiv 13 \mod 15$ so he sends 13 over the airwaves to Alice.

Since the numbers are so small, and exhaustive brute-force search will find Bob's message $m = 7$, but imagine the numbers are hundreds of digits long.

Alice computes $13^d = 13^3 = 169.13 \equiv 4.13 = 52 = 45 + 7 \equiv 7$.

20. issues

We claim that $m^{\phi(pq)} \equiv 1 \mod pq$, with $pq = n$, but that statement above has the proviso that m, n should be relatively prime. The fact (theorem) says nothing about the case if m, n have a common factor greater than 1, so maybe its still true. But actually because $n = pq$, how could m, n have a common factor? Only if Bob was lucky enough to pick at random one of Alice's top secret prime numbers. It is easy for Bob to check if his (randomly chosen message) m is actually a divisor of n, just by dividing. It is (should be) also highly unlikely to happen.

Other issues (to be added) and can be found online, there are various attacks but still RSA is considered very secure under the assumption that it is really difficult to guess what the primes p, q are.

You will hear people talk about *security assumptions* and *one-way functions*. Nobody knows if in mathematics one-way functions exist, but the function of multiplying integers is considered a candidate – its easy for Alice to multiply p, q but very hard for people to reverse that knowing only n (and e).

21. Example

Try this. Alice picks $p = 5, q = 11$ and Bob wants to send 436. [13] Go through the whole protocol. (Find your own e, d values).

Exercise: if you know n and $\phi(n)$ then you can work out p, q.

[13] Hint: this is too big, we can only send $0 < m < n$ to get a unique decryption. Trick: Send it as two two-digit numbers 04 and 36. Alice and Bob would need to agree on this as an extra part of the protocol.

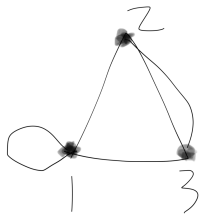
Part 5. Graphs and trees

22. Basic definitions

Definition 22.1 (Graph)**.** A *graph* $G = (V, E)$ is a pair of sets V, E such that each $e \in E$ is associated to some subset $\{v_1, v_2\} \subseteq V$ of size 1 or 2.

The elements of V are called [plural: *vertices*, singular: *vertex*] or *nodes*, and the elements of E are called *edges* or *arcs*.

Example 22.2. If $V = \{1, 2, 3\}, E = \{e_{11}, e_{12}, e_{13}, e_{23}, e_{32}\}$ where e_{ij} is associated to $\{i, j\} \subseteq V$ then $G = (V, E)$ is a graph. We can *visualise* G as a picture, with a dot for each element of V and a line between v_1 and v_2 if $\{v_1, v_2\}$ is associated to some $e \in E$, so in this case we get



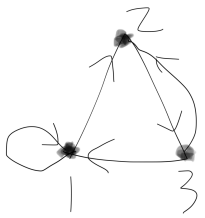
If $\{v_1, v_2\}$ is associated to more than one edge in some graph G, these edges are called *multiple edges* or *multi-edges* and G is said to contain multiple edges. When each subset $\{v_1, v_2\} \subseteq V$ of size 1 or 2 is associated to at most one element of E, we can choose to label each edge by a subset of V of size 1 or 2, and write $E \subseteq \mathscr{P}(V)$. [14]

A *subgraph* $G = (V, E)$ is a graph $G' = (V', E')$ where $V' \subseteq V, E' \subseteq E$.

Definition 22.3 (Directed graph)**.** A *directed* graph $G = (V, E)$ is a pair of sets V, E such that each $e \in E$ is associated to some ordered pair $(v_1, v_2) \in V \times V$. If $(u, v) \in E$ we call $u \in V$ the *source* vertex and v the *terminal* vertex. In this case when we visualise G as a picture, we draw arrows on the edges to indicate their *direction*, from source to terminal.

Example 22.4. If $V = \{1, 2, 3\}, E = \{e_{11}, e_{12}, e_{31}, e_{23}, e_{32}\}$ where e_{ij} is associated to $(i, j) \in V^2$ then $G = (V, E)$ is a directed graph.

[14] Recall from the worksheet the *power set* of a set A is the set $\mathscr{P}(A)$ of all subsets of A.



Again if more than one edge is associated to the same ordered pair, we call them *multi-edges*. If G is directed with no multi-edges we can choose to label E by ordered pairs $V \times V = V^2$, and we write $E \subseteq V^2$. In our example there are no multi-edges, since $(2, 3)$ and $(3, 2)$ are different elements of V^2.

Exercise 22.5. Complete this list of definitions (use the previous 37181 notes Chapter 9, a textbook or online)

(1) loop
(2) multiple edges/multi-edge
(3) simple graph
(4) path
(5) length of a path
(6) circuit
(7) connected
(8) disconnected
(9) simple path
(10) endpoint(s) of an edge
(11) edge incident to a vertex
(12) adjacent vertices
(13) degree of a vertex (for undirected graphs), notation $\deg(v)$
(14) in-degree and out-degree (for directed graphs)
(15) complete graph K_n on n vertices
(16) complete bipartite graph on $m + n$ vertices

Note that all our definitions are given (precisely) in terms of set theory, rather than some picture-description. You should know now that this is important for when it comes time to *proving* facts about graphs. If we have imprecise definitions, we will have trouble in our proofs.

I often use *node* instead of *vertex* to make the English simpler. Remember the grammar for plural versus singular if you use *vertices, vertex*.

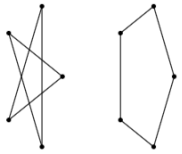
Definition 22.6 (Adjacency matrix)**.** Let $G = (V, E)$ be a graph (directed or undirected). Assume $|V| = n$ and $V = \{1, 2, \ldots, n\}$. The

adjacency matrix for G is a $n \times n$ matrix $A = (a_{ij})$ where a_{ij} is the number of edges from vertex i to vertex j.

Example 22.7. The adjacency matrices for the two examples given above are

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Exercise 22.8. Give the adjacency matrices for these graphs.



Now that we have all these definitions, we can prove some theorems.

Theorem 22.9. *If $G = (V, E)$ is a graph (undirected) and $|E| = n$ then*

$$\sum_{v \in V} \deg(v) = 2n.$$

Do we have our definitions correct? How does degree work with loops – does a loop count 1 or 2 towards the degree? [15]

Proof. Imagine drawing a mark on your picture of G for each pair (v, e) where v is an endpoint of e (and draw it close to v). Then the number of marks you have drawn is $\sum_{v \in V} \deg(v)$. Now if each edge has exactly two marks drawn on it, we have our theorem. So, it is important that each loop has two marks drawn on it, which it does as we have specified the drawing algorithm. □

Proof. A more formal version: $\sum_{v \in V} \deg(v)$ is counted by the number of pairs (v, e) where v is an endpoint of e. Since for each e there are exactly two such pairs (even when it is a loop, we have two pairs (v, e) and (v, e)) so this number is equal to $2n$. □

Exercise 22.10. What would be the corresponding statement for directed graphs? Can you prove it?

Exercise 22.11. How many edges does the complete graph on n vertices have?

[15] In the original 37181 notes this is a problem.

Theorem 22.12. *If $G = (V, E)$ is a graph with $V = \{1, 2, \ldots, s\}$ and $s \times s$ adjacency matrix A, then the number of paths starting at i and ending at j of length $n \geqslant 1$ is the ij-th entry in A^n.*

What technique to prove this? First of all, check its true (do we have our definitions of paths, and length of path, correct?) This is a statement about paths of length $n \in \mathbb{N}_+$ so maybe *induction* is appropriate.

Proof. Let $P(n)$ be the statement that the number of paths starting at i and ending at j of length $n \geqslant 1$ is the ij-th entry in A^n.

Then $P(1)$ is exactly the definition of adjacency matrix: the number of paths of length 1 from i to j is given by a_{ij}.

Assume $P(k)$ is true and consider $A^{k+1} = AA^k$. If we write $A^k = (b_{ij})$ then by assumption b_{ij} is the number of paths from i to j of length k. By the definition of matrix multiplication the ij-th entry of $A^{k+1} = AA^k$ is

$$a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{t=1}^{n} a_{it}b_{tj}$$

which counts paths whose first step goes via some different vertex t, so counts all the paths that start at i and make one step to be at vertex t and then follow a path from t to j. Since all of these paths are different, we get the correct count. □

Exercise 22.13. What would be the corresponding statement for directed graphs? The proof should go through exactly the same.

Application: `https://math.stackexchange.com/questions/92555/counting-the-number-of-paths-on-a-graph`
How many different ways are there to unlock an android phone?

Theorem 22.14. *Every path contains a simple path*

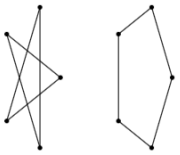
Proof by induction on length of the path.

23. Graph isomorphism

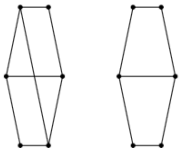
In the next definition, we assume our graphs don't have multi-edges, just to make the statements easier to say.

Definition 23.1. Let $G = (V_1, E_1)$ and $H = (V_2, E_2)$ be two graphs that do not have multi-edges, and so we can assume $E_i \subseteq \mathscr{P}(V_i)$. We say G, H are *isomorphic* if there is a bijection $f : V_1 \to V_2$ such that for all $x, y \in V$, we have $\{x, y\} \in E_1$ if and only if $\{f(x), f(y)\} \in E_2$.

Exercise 23.2. Decide if these two graphs are isomorphic.



Exercise 23.3. Decide if these two graphs are isomorphic.

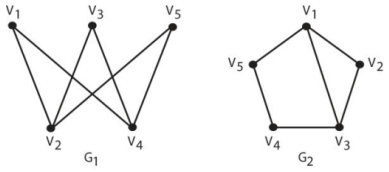


To show non-isomorphic, it is useful to have some *invariants*. For example, if the number of vertices is different, you can say No straight away. What other things might be preserved by an isomorphism?

Exercise 23.4. Decide whether these are invariants of a graph under isomorphism (that is, if you apply an isomorphism map f to G then $f(G)$ has the same number of these things as G does.)

(1) number of loops (at each vertex)
(2) number of vertices of degree d
(3) number of edges
(4) number of cycles of length r

Exercise 23.5. Decide if these two graphs are isomorphic.

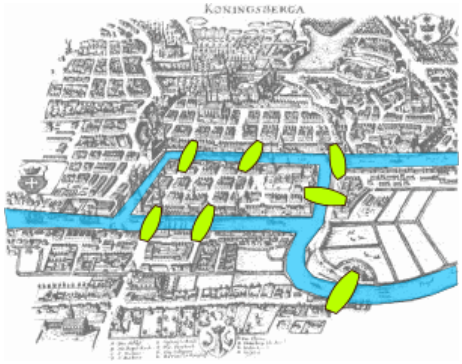


There is some very interesting current research on the complexity of deciding if two input graphs are isomorphic – can it be done in polynomial time, or is it NP-complete?

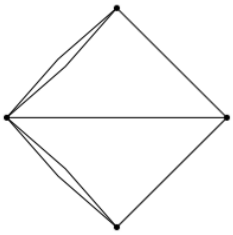
24. Euler paths and circuits

An Euler path in a graph is a path that traverses (uses) every edge exactly once. An Euler circuit in a graph is an Euler path that starts and ends at the same point.

Exercise 24.1. Decide if there is a way to walk around this town crossing every bridge exactly once (and return to your starting point).

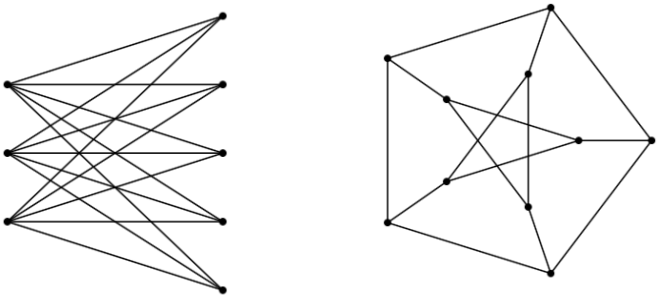


or



[16]

Exercise 24.2. Decide if these graphs have Euler paths or circuits.



It turns out it is easy to decide whether or not a graph has an Euler path or circuit: if a vertex has odd degree, then you cannot cross every edge of the graph without getting stuck at this vertex. This becomes the following theorem.

Theorem 24.3. *A graph $G = (V, E)$ has an Euler circuit if and only if every vertex has even degree. G has an Euler path if either it has exactly 0 or 2 vertices of odd degree.*

This gives us a polynomial time algorithm to decide if G has a circuit/path or not: just compute the degree of each vertex.

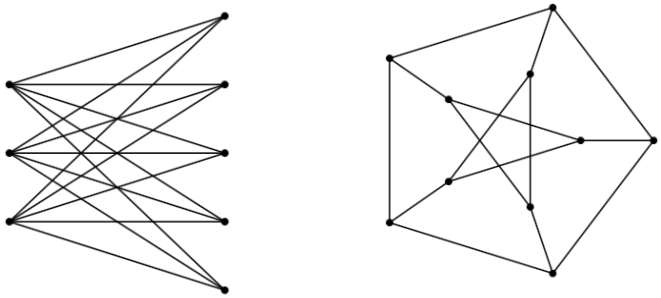
25. Hamiltonian paths and circuits

Instead of visiting every edge, how about a path or circuit that visits every node exactly once. This would be useful if you were a salesperson who needed to visit a whole bunch of cities and didn't want to waste time/money visiting the same city twice.

[16](image by Bogdan Giuşcă - Public domain (PD), based on the image, CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=112920`)

Definition 25.1. A Hamiltonian cycle is a circuit in G that visits every vertex exactly once. A Hamiltonian path is a path in G that visits every vertex exactly once.

Exercise 25.2. Decide if these graphs have Hamiltonian paths or cycles.



The old 37181 notes say it is *unfortunate* that we don't have a lot of results about Hamiltonian cycles. But actually it is interesting: the problem of deciding if G has an Hamiltonian cycle is NP-complete.

The notes give this theorem, with a proof by contradiction.

Theorem 25.3. *Suppose $G = (V, E)$ is a simple graph with $|V| = n \geqslant 3$. Suppose that for every pair of non-adjacent vertices v_1 and v_2 we have $\deg(v_1) + \deg(v_2) \geqslant n$. Then G has a Hamiltonian cycle.*

Note, if a graph is not simple, you could remove all the loops first, then check this theorem. Loops don't help at all with finding a Hamiltonian path of course. If you can think of some clever way to check if a graph fails to have a Hamiltonian cycle (which can be checked in polynomial time in the number of vertices) then you get USD1M from the Clay Institute.

26. Trees

A *tree* is an undirected graph $G = (V, E)$ that satisfies any of the following equivalent conditions: [17]

(1) G is connected and has no cycles.
(2) G has no cycles, and a simple cycle is formed if any edge is added to G.
(3) G is connected, but would become disconnected if any single edge is removed from G.
(4) G is connected and K_3 is not a *minor* of G.

[17](which means, take one as your definition, then prove all the others are equivalent to it)

(5) Any two vertices in G can be connected by a unique simple path.

By convention, we don't allow the empty graph to be a tree. Check the conditions if G was empty.

Exercise 26.1. Draw pictures of trees having $|V| = 1, 2, 3, 4, 5$. How many different trees (up to graph isomorphism) are there of each size. What do you notice about the number of edges versus the number of vertices?

A *leaf* is a vertex of degree 1 in a tree. A *forest* is a graph where each connected component is a tree.

Here is an example of how to show two of the criteria given above are equivalent.

Theorem 26.2. *G is connected and has no cycles if and only if G is connected, but would become disconnected if any single edge is removed from G.*

Note if an only if means we must prove two directions.

Proof. Assume G is connected and has no cycles. Then G is connected. Suppose (for contradiction) some single edge is removed (keeping its endpoints x, y), and G is still connected. Then there is a path in G' from x to y, so taking this path together with the removed edge we have a cycle passing through x, y. Contradiction.

Now assume G is a connected graph with the property that removing a single edge always disconnects it. Then G is connected. Suppose G has a cycle. Then removing an edge on that cycle does not disconnect, contradiction. So G doesn't have any cycles. □

Exercise 26.3. This theorem proves that 1. and 3. are equivalent. Show the rest are equivalent (note you can use *syllogism* and just show, for example, 1. implies 2., 2. implies 4., 4. implies 5., 5. implies 3. and you are done.

Recall a *subgraph* $G = (V, E)$ is a graph $G' = (V', E')$ where $V' \subseteq V, E' \subseteq E$. A *minor* of $G = (V, E)$ means a graph obtained from G by taking a subgraph $G' = (V', E')$ and then replacing paths in G' from x to y where each v_i on the path has degree 2, and replacing the path by an edge. So saying that G contains K_3 as a minor just means you can take a subgraph which is a cycle, keep 3 vertices and replace the paths between them by edges. So 1. and 4. are immediately equivalent.

Theorem 26.4. *A tree with $n \in \mathbb{N}_+$ vertices has $n - 1$ edges*

Proof. Strong induction. True for $n = 1, 2$ (check: there is only one graph of 1 vertex and one with two vertices). Assume true for $k \geqslant 2$ and consider T with $k+1$ vertices. If T has no edges then it is disconnected ($k+1 \geqslant 3$), so T has an edge. Choose one (there are only finitely many to choose from) and erase it, leaving its endpoints, to get two trees (not connected after removing an edge by Theorem 26.2). Each tree has $\leqslant k$ vertices so the statement is true for them, done. □

We mostly restrict to finite graphs and trees in First Year Discrete Maths.But infinite graphs and trees lead to very interesting mathematics.

Theorem 26.5 (König's lemma, special case)**.** *If T is an infinite tree where each vertex has finite degree, then T has an infinite simple path.*

Check Wikipedia for a proof (or do you think is sounds completely obvious?).

According to Wikipedia *This proof is not generally considered to be* constructive*, because at each step it uses a proof by contradiction to establish that there exists an adjacent vertex from which infinitely many other vertices can be reached, and because of the reliance on a weak form of the* axiom of choice*.*

27. Spanning trees

Definition 27.1. A spanning tree of a graph $G = (V, E)$ is a tree $H = (V, E')$ with $E' \subseteq E$.

Note if G has n vertices then a spanning tree (if it exists) must have $n - 1$ vertices. Note that since a tree must be connected, if G is not connected then it cannot have a spanning tree.

Exercise 27.2. Find two non-isomorphic spanning trees for this graph:

Theorem 27.3. *Every connected non-empty finite graph contains a spanning tree.*

Proof. G is non-empty so choose a vertex, v_1 and let $T = v_1$. The the following algorithm terminates in a finite number of steps (since G is finite.)

Loop invariant: T is a tree. True at the start.

While T does not contain all vertices of G, do:

choose a vertex in $G \setminus T$, x. Since G is connected, for each $v \in T$ there is a path from x to v, and out of all these paths we can choose one path that only intersects with T at a single vertex (if not, step back an edge). Add this path to T.

(loop invariant: T is still a tree by our careful choice of path)

Stops because each time T gets bigger and G is finite. At the end, T contains all vertices (if not, the algorithm can keep going) and is (still) a tree. □

In later optimisation courses you will study (more?) *efficient* algorithms to construct spanning trees. Use Big O to make precise how efficient.

28. Rooted trees

A *rooted tree* is just a tree with one vertex nominated to be called the *root*. We often like to draw the root at the top (or bottom) of the picture and then draw vertices connected by an edge to that vertex below (or above), and continue, so it really looks like an upside-down tree.

PIC

28.1. **Application: bracket-free arithmetic.** We can represent expressions like Inorder, preorder, postorder are three different conventions on how to write the nodes of a rooted binary tree as a string.

(a) Inorder (Left, Root, Right)
(b) Preorder (Root, Left, Right)
(c) Postorder (Left, Right, Root)

Useful in computing, efficient to represent (old calculators would use this for display, input); recursively defined.

29. Euler's formula

In class we draw planar graphs on balloons to come up with the following. We defined a *face* to be a polygonal region of the balloon bounded by edges but with no edges inside.

Theorem 29.1. *If G is planar, finite, then $|V| - |E| + |F| = 2$ (for any representation/drawing of G on the plane without edge crossings.)*

Application: K_5 cannot be planar. Proof: add up $|V| - |E| + |F|$. 5,5.4=20, 5-20+F=2, F=17. But each face is a triangle, so ...

References

[1] Y. Moschovakis. *Notes on set theory*. Undergraduate Texts in Mathematics. Springer, New York, second edition, 2006.

[2] M. Sipser. *Introduction to the Theory of Computation*. Cengage Publishing, 3rd edition, 2012.

[3] Wikipedia contributors. Modus tollens — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Modus_tollens&oldid=848273828`, 2018. [Online; accessed 23-July-2018].

[4] Wikipedia contributors. Zermelo–fraenkel set theory — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Zermelo%E2%80%93Fraenkel_set_theory&oldid=851193109`, 2018. [Online; accessed 25-July-2018].

Email address: `murrayelder@gmail.com`