Algorithmic Foundations 2

Section 4 – Sequences, summations, integers & matrices

Dr. Gethin Norman

School of Computing Science
University of Glasgow

Sequences – Example

Consider the sequence $a_1, a_2, a_3, \ldots$ where $a_n = 1/n$

- i.e. the sequence $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$

Consider the sequence $a_0, a_1, a_2, a_3, \ldots$ where $a_n = a+n \cdot d$

- this is an arithmetic progression
- i.e. the sequence $a, a+d, a+2 \cdot d, a+3 \cdot d, \ldots$

Consider the sequence $a_0, a_1, a_2, a_3, \ldots$ where $a_n = a \cdot r^n$

- this is a geometric progression
- i.e. the sequence $a, a \cdot r, a \cdot r^2, a \cdot r^3, \ldots$

Summations and product – Examples

Suppose we have a sequence $a_1, a_2, a_3, \ldots$

$$\sum_{i=m}^{n} a_i = a_m + a_{m+1} + a_{m+2} + \cdots + a_{n-1} + a_n$$

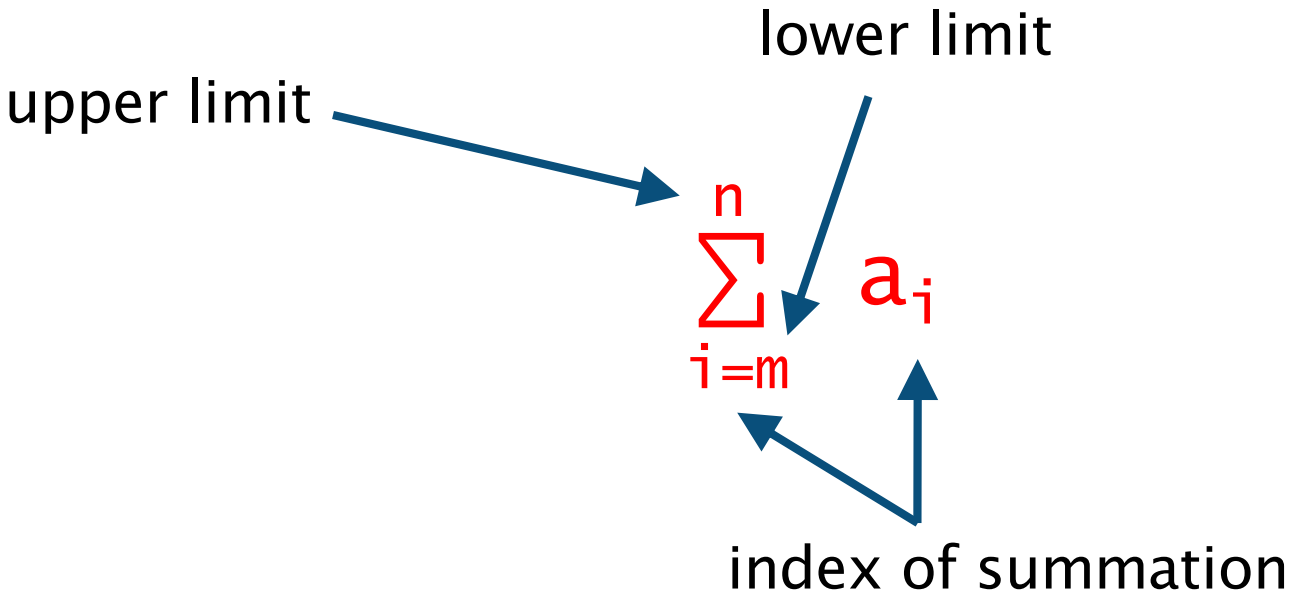
$$\prod_{i=m}^{n} a_i = a_m \cdot a_{m+1} \cdot a_{m+2} \cdots a_{n-1} \cdot a_n$$

"Sigma" for sum and "Pi" for product

Summations – Notation

Suppose we have a sequence $a_1, a_2, a_3, \ldots$

$$\sum_{i=m}^{n} a_i = \sum_{j=m}^{n} a_j = \sum_{k=m}^{n} a_k$$



Summations – Examples

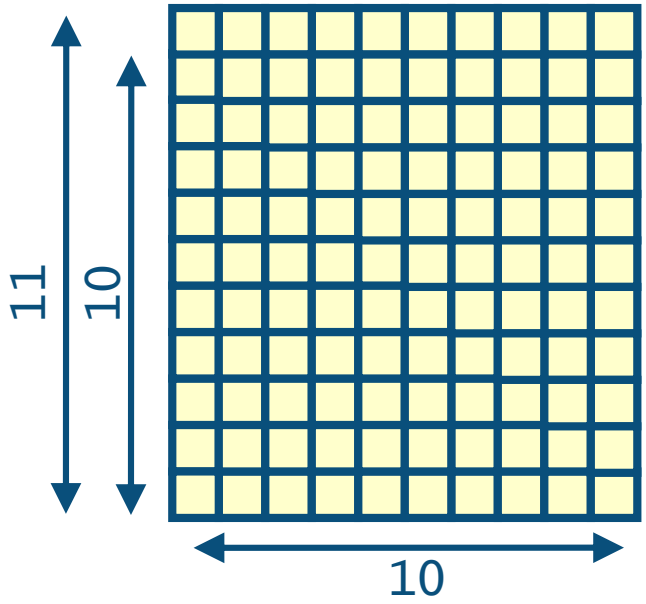
The sum of the first hundred positive integers

$$\sum_{i=1}^{100} i = 1 + 2 + 3 + \cdots + 99 + 100$$

What is the answer?

Summations – Examples

$$\sum_{i=1}^{10} i = 1 + 2 + 3 + \cdots + 9 + 10$$



We now have a rectangle of size 10 by 11

- therefore area equals $10 \cdot 11 = 110$ units
- we need half of that, i.e. summation equals $110/2$

In general for a summation up to n have a square of size n by (n+1)

- therefore area equals $n \cdot (n+1)$ units
- we need half of that, i.e. summation equals $n \cdot (n+1)/2$

$$\sum_{i=1}^{n} i = n \cdot (n+1)/2$$

Summations – Examples

$$\sum_{i=1}^{n} i = n \cdot (n+1)/2$$

Therefore

- sum of the first 10 positive integers equals 55
- sum of the first 100 positive integers equals 5,050
- sum of the first 1,000 positive integers equals 500,500
- sum of the first 1,000,000 positive integers equals 500,000,500,000

Summations – A challenge

What is the following summation?

$$\sum_{i=50}^{100} i = 50 + 51 + 52 + \cdots + 99 + 100$$

$$= \sum_{i=1}^{100} i - \sum_{i=1}^{49} i$$

$$= 100 \cdot 101/2 - 49 \cdot 50/2 = 3825$$

$$\sum_{i=1}^{n} i = n \cdot (n+1)/2$$

Summations – Shifting indices

$$\sum_{i=1}^{n} i^2 = \sum_{j=0}^{n-1} (j+1)^2$$

If we replace i by $j+1$

- lower limit equals $i=1$ substituting we get $j+1=1$ yielding $j=1-1=0$
- upper limit equals $i=n$ substituting we get $j+1=n$ yielding $j=n-1$
- summand equals i^2 substituting we get $(j+1)^2$

Summations – a theorem

If **r=1**, then

$$\sum_{i=0}^{n} r^i = (n+1)$$

If **r≠1**, then

$$\sum_{i=0}^{n} r^i = (r^{n+1}-1)/(r-1)$$

You can prove this for yourself once we have covered induction

Summations and sets

If $S = \{s_1, s_2, s_3, \ldots, s_n\}$, then

$$\sum_{i \in S} a_i = a_{s1} + a_{s2} + a_{s3} + \cdots + a_{sn}$$

Summations – Rearranging

$a + a = 2 \cdot a$

$$\sum_{i=1}^{n} c = n \cdot c$$

$$\sum_{i=0}^{n} c = (n+1) \cdot c$$

$(a_1 + b_1) + (a_2 + b_2) = (a_1 + a_2) + (b_1 + b_2)$

$$\sum_{i=0}^{n} (a_i + b_i) = \sum_{i=0}^{n} a_i + \sum_{i=0}^{n} b_i$$

Summations – Rearranging

$$c \cdot a_1 + c \cdot a_2 = c \cdot (a_1 + a_2)$$

$$\sum_{i=0}^{n} c \cdot a_i = \sum_{i=0}^{n} c \cdot a_i$$

$$(a_{1,1} + a_{1,2}) + (a_{2,1} + a_{2,2}) = (a_{1,1} + a_{2,1}) + (a_{1,2} + a_{2,2})$$

$$\sum_{i=0}^{m} \sum_{j=0}^{n} a_{i,j} = \sum_{j=0}^{n} \sum_{i=0}^{m} a_{i,j}$$

Summations – Rearranging

What you cannot do...

$(a_1 \cdot b_1) + (a_2 \cdot b_2) \neq (a_1 + a_2) \cdot (b_1 + b_2)$

- since rhs equals $(a_1 \cdot b_1) + (a_1 \cdot b_2) + (a_2 \cdot b_1) + (a_2 \cdot b_2)$

$$\sum_{i=0}^{m} \sum_{j=0}^{n} (a_i \cdot b_j) \neq \sum_{i=0}^{m} a_i \cdot \sum_{j=0}^{n} b_j$$

Summations – Rearranging

$$\sum_{i=0}^{n} (a+i\cdot d) = \sum_{i=0}^{n} a + \sum_{i=0}^{n} d\cdot i$$

$$= \sum_{i=0}^{n} a + d\cdot \sum_{i=0}^{n} i$$

$$= (n+1)\cdot a + d\cdot(n+1)\cdot n/2$$

Division

Given integers **a** and **b**

- a divides b if a is not zero and there is an integer c such that $a \cdot c = b$
- "a is a factor of b"
- "b is a multiple of a"

We say "**a|b**" when **a** divides **b** and in predicate logic:

$$\forall a \in \mathbb{Z}.\forall b \in \mathbb{Z}.(a|b \rightarrow ((a \neq 0) \wedge \exists c \in \mathbb{Z}.(a \cdot c = b)))$$

Division

If $a|b$ and $a|c$, then $a|(b+c)$

- if a divides b and a divides c, then a divides b plus c

Proof.

- if $a|b$, then $a \neq 0$ and there exists $x \in \mathbb{Z}$ such that $a \cdot x = b$
- if $a|c$, then $a \neq 0$ and there exists $y \in \mathbb{Z}$ such that $a \cdot y = c$
- combining these facts we have
- $a \neq 0$ and $b+c = a \cdot x + a \cdot y = a \cdot (x+y)$
- and hence $a|(b+c)$

In predicate logic $\forall a \in \mathbb{Z}. \forall b \in \mathbb{Z}. \forall c \in \mathbb{Z}. (a|b \wedge a|c) \rightarrow (a|(b+c))$

Division

Similarly we can show…

If $a|b$, then $a|(b \cdot c)$ for all c

If $a|b$ and $b|c$, then $a|c$

The division algorithm

If a is an integer and d is a positive integer, then there exists unique integers q and r such that $0 \leq r < d$ and $a = d \cdot q + r$

- a divided by d equals q remainder r
- a is the dividend
- d is the divisor
- q is the quotient
- r is the remainder
- the remainder is non-negative and less than the divisor
  - required for uniqueness

$$\forall a \in \mathbb{Z}.\forall d \in \mathbb{Z}^+.\exists! q \in \mathbb{Z}.\exists! r \in \mathbb{Z}^+.((0 \leq r < d) \wedge (a = d \cdot q + r))$$

- $\exists!$ "exists a unique"

Primes

An integer **p>1** is **prime** if the only positive factors are **1** and **p**

- if p is not prime it is called composite

The fundamental theorem of arithmetic:

- every positive integer (≥1) can be expressed as a unique product of primes

$$n = \prod_{i=1}^{k} p_i^{e_i} = p_1 \cdots p_1 \cdot p_2 \cdots p_2 \cdots p_k \cdots p_k$$

Examples

- $60 = 2^2 \cdot 3 \cdot 5 = 2 \cdot 2 \cdot 3 \cdot 5$
- $100 = 2^2 \cdot 5^2 = 2 \cdot 2 \cdot 5 \cdot 5 = 2 \cdot 2 \cdot 1 \cdot 5 \cdot 5 = 2^2 \cdot 3^0 \cdot 5^2$
- unique means there is no other factoring into primes

Fundamental Theorem of Arithmetic

The fundamental theorem of arithmetic:

- every positive integer (≥1) can be expressed as a unique product of primes

Proof (not uniqueness).

If **n** is prime we are done, assume **n** is composite

- n has a positive divisor $1<p<n$ and let p_1 be the smallest of these divisors
- p_1 is prime
  - if p_1 was not prime there exists k such that $1<k<p_1$ and k divides p_1
  - then since p_1 is a divisor of n we have k is also a divisor of n
  - this contradicts the fact that p_1 is the smallest divisor of n

Fundamental Theorem of Arithmetic

The fundamental theorem of arithmetic:

- every positive integer (≥1) can be expressed as a unique product of primes

Proof (not uniqueness).

If n is prime we are done, assume n is composite

- n has a positive divisor $1<p<n$ and let p_1 be the smallest of these divisors
- p_1 is prime
- hence we have $n = n_1 \cdot p_1$ where p_1 is prime and $n_1<n$

Now repeat the argument with n_1, i.e. if n_1 is prime we are done

- otherwise $n_1 = n_2 \cdot p_2$ where p_2 is prime and $n_2<n_1$ and $p_2 \geq p_1$

… this process terminates due to the Well Ordering Principle (WOP)

- WOP: every non-empty set of positive integers has a least element

Primes

A number is composite if it is not a prime

The hard way to test if **n** is prime

- if n is divisible by 2 **return** (“composite”)
- if n is divisible by 3 **return** (“composite”)
- if n is divisible by 4 **return** (“composite”)

  …
- if n is divisible by n-1 **return** (“composite”)
- **return** (“prime”)

Question: is **n>2** ever divisible by **n-1**?

Primes

If n is composite it has a prime divisor $\leq\sqrt{n}$

Proof.

- if n is composite then $n=a\cdot b$ for some a and b
- hence either $a\leq\sqrt{n}$ or $b\leq\sqrt{n}$
  - otherwise $a>\sqrt{n}$ and $b>\sqrt{n}$ and therefore $a\cdot b > \sqrt{n}\cdot\sqrt{n} = n$
- without loss of generality suppose $a\leq\sqrt{n}$
- either a is prime or due to the fundamental theorem of arithmetic, can be expressed as a product of primes
- in either case we have found a prime divisor $\leq\sqrt{n}$

Primes

If n is composite it has a prime divisor $\leq\sqrt{n}$

We now have a simpler test for primality

- if a number n is prime, then it is **not** composite
- if a number n is prime, then it does **not** have a prime divisor $\leq\sqrt{n}$
- therefore we can test if n is divisible by primes in the range 2 to $\sqrt{n}$
- if none are found n must be prime

Example: is 41 prime?

- $\sqrt{41} = 6.4031242\ldots$ the only primes not exceeding $\sqrt{41}$ are 2, 3, and 5
- none of these divides 41, therefore 41 is prime

Example: is 51 prime?

- $\sqrt{51} = 7.141428\ldots$ the only primes not exceeding $\sqrt{51}$ are 2, 3, 5 and 7
- 3 divides 51, therefore 51 is composite

Prime factorisation

Find the prime factorisation of **63336**

2 is the first prime

- 63336/2 = 31668
- 31668/2 = 15834
- 15834/2 = 7917

  (2 does not divide 7917 so move onto next prime)
- 7917/3 = 2639

  (3 does not divide 2639 so move onto next prime)

  (5 does not divide 2639 so move onto next prime)
- 2639/7 = 377

  (7 does not divide 377 so move onto next prime)

  (11 does not divide 377 so move onto next prime)
- 377/13 = 29

  29 prime (since $\sqrt{29}=5.196...$ and not divisible by any prime ≤ 13)

So $63336 = 2^3 \cdot 3 \cdot 7 \cdot 13 \cdot 29$

Prime factorisation

Assume

- **nextPrime(i)** delivers next prime number greater than **i**
  - e.g. **nextPrime(7)=11** and **nextPrime(nextPrime(7))=13**
- **floor(sqrt(n))** delivers largest integer less than the square root of **n**
  - e.g. **floor(sqrt(97))=9**

```
p:=2; // the first prime
rootN := floor(sqrt(N)) // where we stop the search for primes
while (p ≤ rootN) // still primes to explore
  if p|N then // p is a prime divisor
    print(p); // state this fact
    N:=N/p; // remove the divisor from N
    rootN:=floor(sqrt(N)); // recalculate where we stop
  else
    p := nextPrime(p); // not a divisor - move to next prime
print(N); // final prime when we exit while loop
```

Prime factorisation

- N=7007, p=2 and rootN=83
  - p=**nextPrime**(2)=3
  - p=**nextPrime**(3)=5
  - p=**nextPrime**(5)=7
  - **print**(7)
  - N= 7007/7 = 1001
  - rootN = **floor**(**sqrt**(1001)=31
  - **print**(7)
  - N = 1001/7 = 143
  - rootN = **floor**(**sqrt**(143)=11
  - p=**nextPrime**(7)=11
  - **print**(11)
  - N = 143/11 = 13
  - rootN = **floor**(**sqrt**(13)=3
  - **print**(13)

```
p := 2;
rootN := floor(sqrt(N))
while p≤rootN
  if p|N then
    print(p);
    N := N/p;
    rootN := floor(sqrt(N));
  else
    p := nextPrime(p);
print(N);
```

7007 = 7·7·11·13

Testing Primality

How easy is it to test a number for primality?

- say a number n with 100 decimal digits
- $\sqrt{n}$ will have about 50 decimal digits
- these means about 10^{50} possible divisors
- so infeasible to test all possible divisors

How many prime numbers are there?

Assume there are finitely many prime numbers, i.e. $p_1, p_2, \ldots, p_n$

- let p_n be the greatest prime
- multiply all the primes together add 1 call this n

  i.e. $n = p_1 \cdot p_2 \cdots p_n + 1$
  - n is not divisible by $p_1=2$, we get remainder 1
  - n is not divisible by $p_2=3$, we get remainder 1
  - n is not divisible by $p_3=5$, we get remainder 1
  - …
  - n is not divisible by p_n, we get remainder 1
- by the FTA we know n has a prime factorisation
- therefore n must have a prime divisor greater than p_n
- yielding a contradiction, hence there cannot be a greatest prime
- i.e. there are an infinite number of primes

Outline of Section 4

Sequences and Summations

Divisibility and the division algorithm

Primes and Fundamental theorem of arithmetic (FTA)

Greatest common divisor

Mod arithmetic

Euclidean algorithm for gcd

Bases

Matrices

Greatest common divisor

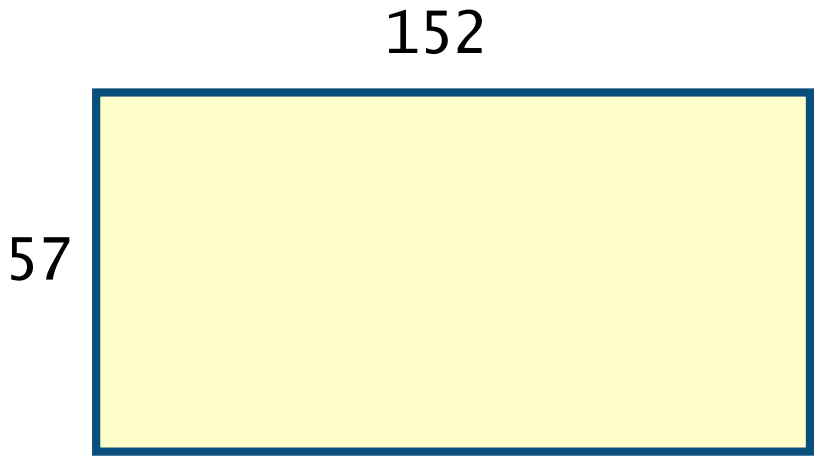
The greatest common divisor

- **gcd**(a,b) is largest d such that d|a and d|b
- if **gcd**(a,b) = 1 then a and b are relative primes

Finding **gcd**'s efficiently is a small but essential component of RSA

A geometric view:

what is the largest size of square tile that can be used to tile a rectangular area (without using a tile cutter)?



Greatest common divisor

The greatest common divisor

- **gcd**(a,b) is largest d such that d|a and d|b
- if **gcd**(a,b) = 1 then a and b are relative primes

Two naïve algorithms for **gcd(a,b)**

- start with x at 1 up to **min**(a,b) testing if x|a and x|b
  and remember the last (largest) successful value
- start with x at **min**(a,b) and count down to 1 testing if x|a and x|b
  stop when the first value of x is found

Greatest common divisor

The greatest common divisor

- **gcd**(a,b) is largest d such that d|a and d|b
- if **gcd**(a,b) = 1 then a and b are relative primes

Another naïve algorithm for **gcd**

- based on prime factorisation of a and of b
- suppose prime factorisation of a and b are given by

  $a = p_1^{a1} \cdot p_2^{a2} \cdot p_3^{a3} \cdots p_n^{an}$ and $b = p_1^{b1} \cdot p_2^{b2} \cdot p_3^{b3} \cdots p_n^{bn}$

  then **gcd**(a,b) $= p_1^{\min(a1,b1)} \cdot p_2^{\min(a2,b2)} \cdot p_3^{\min(a3,b3)} \cdots p_n^{\min(an,bn)}$

Example:

- prime factorisation of 120 is $2 \cdot 2 \cdot 2 \cdot 3 \cdot 5 = 2^3 \cdot 3^1 \cdot 5^1$
- prime factorisation of 500 is $2 \cdot 2 \cdot 5 \cdot 5 \cdot 5 = 2^2 \cdot 3^0 \cdot 5^3$
- **gcd**(120,500) $= 2^2 \cdot 3^0 \cdot 5^1 = 20$

Least common multiple

The least common multiple

- **lcm**(a,b) is the smallest/least x such that a|x and b|x

A naïve algorithm based on prime factorisation of **a** and of **b**

- suppose prime factorisation of a and b are given by
  $a = p_1^{a1} \cdot p_2^{a2} \cdot p_3^{a3} \cdots p_n^{an}$ and $b = p_1^{b1} \cdot p_2^{b2} \cdot p_3^{b3} \cdots p_n^{bn}$
- then **lcm**(a,b) $= p_1^{\max(a1,b1)} \cdot p_2^{\max(a2,b2)} \cdot p_3^{\max(a3,b3)} \cdots p_n^{\max(an,bn)}$

Example:

- prime factorisation of 95256 is $2 \cdot 2 \cdot 2 \cdot 3 \cdot 3 \cdot 3 \cdot 3 \cdot 3 \cdot 7 \cdot 7 = 2^3 \cdot 3^5 \cdot 5^0 \cdot 7^2$
- prime factorisation of 432 is $2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 \cdot 3 \cdot 3 = 2^4 \cdot 3^3 \cdot 5^0 \cdot 7^0$
- **lcm**(95256,432) $= 2^4 \cdot 3^5 \cdot 5^0 \cdot 7^2 = 190512$

mod arithmetic

Modulo arithmetic

- `a mod m` is the remainder of `a` after dividing by `m`
- i.e. `a mod m` is the integer `r` such that `a = q·m + r` and `0≤r<m`

Examples

- `17 mod 3 = 2` since `17 = 5·3 + 2`
- `17 mod 12 = 5` (5pm) since `17 = 1·12 + 5`
- `-17 mod 3 = 1` since `-17 = (-6)·3 + 1`

mod arithmetic

We say **a** is congruent to **b** modulo **m** if **m** divides **a-b**

- written $a \equiv b \pmod{m}$
- note it is symmetric if m divides $a-b$, then m also divides $b-a = -(a-b)$

Examples

- $6 \equiv 11 \pmod{5}$ since $6 - 11 = -5$
- $4 \equiv -17 \pmod{3}$ since $4 - (-17) = 21$
- $3 \equiv 5 \pmod{2}$, $3 \equiv 7 \pmod{2}$, $3 \equiv 9 \pmod{2}$, $3 \equiv 11 \pmod{2}$, …

Congruences

Some results...

a is congruent to **b** modulo **m** if and only if **a mod m = b mod m**

If **$a \equiv b \pmod{m}$** and **$c \equiv d \pmod{m}$**, then **$(a+c) \equiv (b+d) \pmod{m}$**

If **$a \equiv b \pmod{m}$** and **$c \equiv d \pmod{m}$**, then **$(a \cdot c) \equiv (b \cdot d) \pmod{m}$**

Congruences

If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $(a+c) \equiv (b+d) \pmod{m}$

Proof.

- by definition $m \mid (a-b)$ and $m \mid (c-d)$
- from results about divisibility (if $m \mid x$ and $m \mid y$, then $m \mid (x+y)$)
- $m \mid ((a-b)+(c-d))$ which rearranging yields $m \mid ((a+c)-(b+d))$
- therefore by definition we have $(a+c) \equiv (b+d) \pmod{m}$ as required

Congruences

If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $(a \cdot c) \equiv (b \cdot d) \pmod{m}$

Proof.

- by definition $a - b = q_1 \cdot m$ and $c - d = q_2 \cdot m$ for some q_1 and q_2
- rearranging we have $a = q_1 \cdot m + b$ and $c = q_2 \cdot m + d$
- therefore $a \cdot c = (q_1 \cdot m + b) \cdot (q_2 \cdot m + d)$
- which rearranging gives $a \cdot c - b \cdot d = m \cdot (d \cdot q_1 + b \cdot q_2 + q_1 \cdot q_2 \cdot m)$
- i.e. $m \mid (a \cdot c - b \cdot d)$
- hence by definition $(a \cdot c) \equiv (b \cdot d) \pmod{m}$ as required

Detour – Public key cryptography

Public key cryptography

- encryption and decryption are carried out using two different keys
- `public key` (available to everyone)
- `private key` (as the name suggests private to a party)

When **`Party A`** wants to communicate confidentially with **`party B`**

- encrypt using `B`'s `public key`
- only decipherable by `B` as only `B` has access to their `private key`

When **`Party A`** wants to send an authenticated message to **`party B`**

- encrypt using `A`'s `private key`
- message is only be decipherable with `A`'s `public key`
- establishes the authenticity of the message

Detour – Public key cryptography

The RSA (Rivest, Shamir and Adleman) public-key encryption algorithm is the basis of most modern secure communications

- the security relies on the fact that it is (or appears to be) computationally infeasible to factorise very large numbers

Although it is possible to test very large numbers for primality

- this is not by testing all the feasible divisors
- indeed the RSA system depends on being able to find and use very large prime numbers

No one knows a feasible way to find the factors of very large numbers

Detour – Public key cryptography

The RSA (Rivest, Shamir and Adleman) public-key encryption algorithm

- each party has a modulus $n=p\cdot q$ where p and q are large primes and an exponent e that is relative prime to $(p-1)\cdot(q-1)$
  - primes p and q can be found relatively quickly
- messages are translated into sequences of integers and integers grouped to form larger integers
- encryption transforms an integer M into an integer $C = M^e \bmod n$
- from C can retrieve M quickly using the fact $C^d \equiv M \pmod{p\cdot q}$ where d is a multiplicative inverse of e modulo $(p-1)\cdot(q-1)$
  - i.e. d is such that $d\cdot e \equiv 1 \pmod{(p-1)\cdot(q-1)}$
- which is easy to find as long as we know p and q
- so need to be able to factorise n
- however $n = p\cdot q$ cannot be factored in a reasonable time

Euclidean algorithm

Euclid (325–265 B.C.)

- introduced an algorithm for finding the greatest common divisor
- used in RSA algorithm (for finding multiplicative inverses)
- relies on the following result

If $a=b \cdot q+r$ where a, b, q & r are integers, then $\gcd(a,b) = \gcd(b,r)$

Euclidean algorithm

If $a=b\cdot q+r$ where a, b, q & r are integers, then $\gcd(a,b) = \gcd(b,r)$

Proof (sketch).

- if we can show that common divisors of a and b and of b and r are the same, then it follows that the greatest must also the same
- therefore sufficient to proof this fact, i.e.
  - if $d \mid a$ and $d \mid b$ for some d, then $d \mid r$
  - if $d \mid b$ and $d \mid r$ for some d, then $d \mid a$

Euclidean algorithm

Show any common divisor of **b** and **r** is a divisor of **a**

If $a = b \cdot q + r$, $d \mid a$ and $d \mid b$ for some d, then $d \mid r$

- rearranging we have $r = a - b \cdot q$
- since $d \mid b$ it follows $d \mid (b \cdot q)$ (we saw this earlier)
- since $d \mid a$ and $d \mid (b \cdot q)$ we have $d \mid (a - b \cdot q)$ (we saw this earlier)
- that is $d \mid r$ as required

If $a = b \cdot q + r$, $d \mid b$ and $d \mid r$ for some d, then $d \mid a$

- since $d \mid b$ it follows $d \mid (b \cdot q)$ (we saw this earlier)
- since $d \mid (b \cdot q)$ and $d \mid r$ we have $d \mid (b \cdot q + r)$ (we saw this earlier)
- that is $d \mid a$ as required

Euclidean algorithm – Recursive formulation

Euclid (325–265 B.C.)

- introduced an algorithm for finding the greatest common divisor

```
gcd(a,b) // assumes a≥b
if (b=0) then
    return a; // gcd of 0 and a is a
else
    r = a mod b;
    gcd(b,r); // replace a with r
```

- if $a = b \cdot q + r$ where a, b, q & r are integers, then $\gcd(a,b) = \gcd(b,r)$
- note we have $b > r \geq 0$ since $r = a \bmod b$

The recursion must end since each call of gcd has a strictly smaller non-negative value for the second parameter than previously

- more on recursion later in the course

Euclidean algorithm – Example

Find gcd(662,414)

- 662 = 414·1 + 248

Find gcd(414,248)

- 414 = 248·1 + 166

Find gcd(248,166)

- 248 = 166·1 + 82

Find gcd(166,82)

- 166 = 82·2 + 2

Find gcd(82,2)

- 82 = 2·41 + 0

Therefore gcd(622,414)=2

```
gcd(a,b) // assumes a≥b
if (b=0) then
    return a; // gcd of 0 and a is a
else
    r = a mod b;
    gcd(b,r); // replace a with r
```

Matrices – Introduction

Many applications, including:

- solving linear equation systems
- computer graphics
- image Processing
- models within computational science and engineering
- quantum computing (and mechanics)
- …

Matrices – Introduction

A matrix is a rectangular array, possibly of numbers

- the rows in a matrix are usually indexed `1` to `m` (from top to bottom)
- the columns in matrix are usually indexed `1` to `n` (from left to right)
- elements are indexed by pairs `(i,j)`, `i` is the row and `j` the column

$$A = [a_{i,j}] = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

Matrices as Functions

An $m \times n$ matrix $A = [a_{i,j}]$ of members of a set S can be encoded as a function

$$f_A : (\{1,\ldots,m\} \times \{1,\ldots,n\}) \rightarrow S$$

such that $f_A(i,j) = a_{i,j}$

Matrices – Summation

The sum **A+B** of two matrices **A** and **B**

- which must have the same number of rows and columns

is the matrix (also with the same number of rows and columns) given by adding corresponding elements, i.e. $A+B = [a_{i,j}+b_{i,j}]$

$$\begin{bmatrix} 2 & 6 \\ 0 & -8 \end{bmatrix} + \begin{bmatrix} 9 & 3 \\ -11 & 3 \end{bmatrix} = \begin{bmatrix} 2+9 & 6+3 \\ 0+(-11) & -8+3 \end{bmatrix} = \begin{bmatrix} 11 & 9 \\ -11 & -5 \end{bmatrix}$$

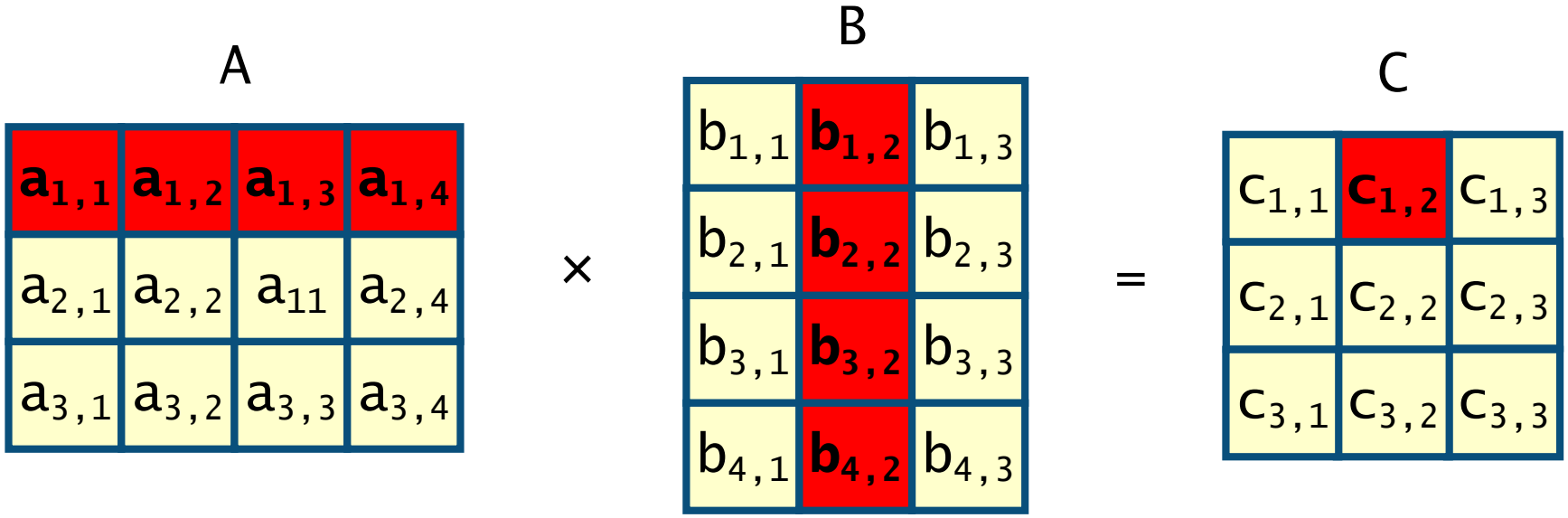
Matrices – Multiplication

For an m×k matrix **A** and a k×n matrix **B**, the product **A×B** is given by the m×n matrix:

$$A \times B = C = [c_{i,j}] \equiv \left[\sum_{r=1}^{k} a_{i,r} \cdot b_{r,j} \right]$$

The element **(i,j)** of **A×B** is given by the vector product of the **i**th row of **A** and the **j**th column of **B** (considered as vectors)

Matrices – Multiplication



$$c_{1,2} = a_{1,1} \cdot b_{1,2} + a_{1,2} \cdot b_{2,2} + a_{1,3} \cdot b_{3,2} + a_{1,4} \cdot b_{4,2}$$

Matrices – Multiplication is not commutative

Note: Matrix multiplication is not commutative

- i.e $A \times B$ and $B \times A$ need to be equal
- one (or both) might not be defined
- but even when both are defined they can be different

Example: if $A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$, then

$$A \times B = \begin{bmatrix} 3 & 2 \\ 5 & 3 \end{bmatrix} \qquad B \times A = \begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix}$$

Matrices – Multiplication is associative

For all matrices A, B and C of size $m \times k$, $k \times l$ and $l \times n$ respectively we have $A \times (B \times C) = (A \times B) \times C$

Proof is available on moodle

We also have distributive properties:

$$A \times (B+C) = (A \times B)+(A \times C)$$

$$(A+B) \times C = (A \times C)+(B \times C)$$

- proofs are question in the tutorials

Matrices – Identities

The identity matrix of order **n**, I_n, is the **n × n** matrix with **1**'s along the upper-left to lower-right diagonal and **0**'s everywhere else

$$I_n = \left[a_{i,j} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases} \right] = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

For any **n × n** matrix **A** we have $A = A \times I_n = I_n \times A$

Matrices – Inverses

For some (but not all) square matrices **A**, there exists a unique multiplicative inverse $\mathbf{A}^{-1}$ of **A**

- if the inverse exists, it is unique and $A^{-1} \times A = A \times A^{-1} = I_n$

We will not go into algorithms for inverting matrices

Matrices – Transpose

If $A=[a_{i,j}]$ is an $m \times n$ matrix, the transpose of A

- often written A^t or A^T

is the $n \times m$ matrix given by $A^t = B = [b_{i,j}] = [a_{j,i}]$

- i.e. we interchange the rows and the columns

Example

$$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \qquad A^t = \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}$$

Matrices – Transpose

If $A=[a_{i,j}]$ is an $m \times n$ matrix, the transpose of A

- often written A^t or A^T

is the $n \times m$ matrix given by $A^t = B = [b_{i,j}] = [a_{j,i}]$

- i.e. we interchange the rows and the columns

A square matrix A is symmetric if and only if $A=A^t$

- that is $a_{i,j} = a_{j,i}$ for all i and j