

## GAC024 Discrete Mathematics

GAC024-aligned discrete mathematics practice; the centre's brief sets the actual assessment.

## Explicit rules for discrete structures

A repeated set member is counted once, a binary carry may exceed a fixed width, and the fewest-edge route may not have the smallest weight.

## State the conventions before solving

- State the universe, representation width, allowed inputs or graph assumptions before solving.
- Show enough working for another reader to reproduce the result.
- Distinguish a mathematical model from its real implementation.

# Sets, unions, complements

| Term                   | Meaning                                                              |
|------------------------|----------------------------------------------------------------------|
| A set                  | A collection of distinct objects; order and repetition do not matter |
| Union and intersection | Everything in either set; what is in both                            |
| Complement             | Everything in the stated universe but outside the set                |
| A subset               | All its elements inside another set                                  |

# Relations, functions, inverses

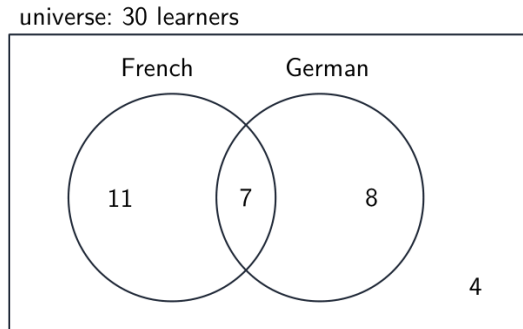
- A relation pairs elements of two sets.
- A function is a relation where each input in its stated domain has exactly one output.
- Different inputs may share an output.
- An inverse relation is a function only when outputs uniquely identify their inputs.

## Inclusion-exclusion subtracts the overlap once

The inclusion-exclusion principle: the size of a union is the sum of the sizes, minus the twice-counted overlap.

$$|A \cup B| = |A| + |B| - |A \cap B|$$

## Worked example: French, German, both, neither



$$|F \cup G| = 18 + 15 - 7 = 26;$$

- Known: 30 learners; 18 study French, 15 German, 7 both.
- Union:  $18 + 15 - 7 = 26$ ; neither:  $30 - 26 = 4$ .
- French-only 11, German-only 8: the four regions sum to 30.

$$N_{\text{neither}} = 30 - 26 = 4$$

## Your turn: the second subtraction

### Your turn

25 students; 14 play football, 12 basketball, 5 both. How many play neither?

- A. 4, from  $25 - (14 + 12 - 5)$
- B. 6, from  $25 - (14 + 12)$
- C. 5, the number who play both
- D. 1, from  $25 - 14 - 12 + 5 - 3$

## Bases, digits, place weights

- A positional number base  $b$  uses digits from zero to  $b$  minus one, with place weights  $b$  to the  $i$ .
- Decimal uses ten; binary two; hexadecimal sixteen.
- Every digit's value is its place value: in binary the places are 1, 2, 4, 8, 16 and so on.
- One hex digit is exactly four bits, so conversion groups a stated width into four-bit blocks.

## Fixed width and wraparound

- For  $n$  unsigned bits, values run from zero to  $2^n - 1$ .
- Distinguish an unrestricted sum from a stored fixed-width result.
- A wraparound rule, if explicitly given, keeps the low  $n$  bits.

$$0 \leq V \leq 2^n - 1, \text{ for } n \text{ unsigned bits}$$

## Worked example: reading 1101 in binary

|   |   |   |   |
|---|---|---|---|
| 8 | 4 | 2 | 1 |
| 1 | 1 | 0 | 1 |

place weights above binary digits

- Known numeral 1101 base 2; weights from the right: 1, 2, 4, 8.
- $V = 1(8) + 1(4) + 0(2) + 1(1) = 13$ .
- The same value is D in hexadecimal.

$$V = 1(8) + 1(4) + 0(2) + 1(1) = 13$$

## Your turn: another numeral

### Your turn

What is the decimal value of 1011 base 2?

- A. 11, from  $8 + 0 + 2 + 1$
- B. 13, from  $8 + 4 + 0 + 1$
- C. 9, from  $8 + 1$
- D. B, its hex digit

# Binary arithmetic and Boolean algebra

- Binary arithmetic adds like decimal, carrying at 2 instead of at 10.
- A bit is one binary digit; a byte is eight.
- Boolean algebra works on true and false with AND, OR and NOT.
- Use inclusive OR and explicit brackets.

$$\neg(A \wedge B) = (\neg A) \vee (\neg B)$$

## Truth tables prove finite equivalence

- A truth table lists every Boolean input combination and its output.
- Matching every row proves equivalence for the same finite Boolean inputs.
- It does not prove physical circuit timing or real-system security.
- Logic gates implement stated operations; a logic circuit connects them.

## Worked example: NOT of an OR



- Known:  $Y = \text{NOT}(A \text{ OR } B)$ ; inputs A and B enter the OR block, then the NOT block.
- Inclusive OR is false only when both inputs are false.
- Row order (0,0), (0,1), (1,0), (1,1): the output column is 1, 0, 0, 0.

## Your turn: one row of the table

### Your turn

For  $Y = \text{NOT}(A \text{ OR } B)$  with  $A = 1$  and  $B = 0$ , what is  $Y$ ?

- A. 0, because  $1 \text{ OR } 0$  is 1 and NOT 1 is 0
- B. 1, because one input is 0
- C. 1, because OR needs both inputs true
- D. 0, because NOT applies only to A

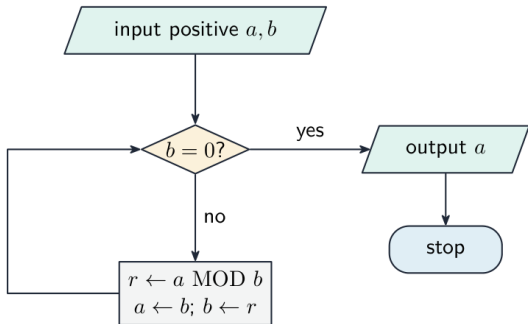
# Algorithms, flowcharts, pseudocode

- An algorithm describes unambiguous steps for a task, and must terminate with the required result for its allowed inputs.
- A flowchart draws it: a decision is a diamond, a process a rectangle.
- Pseudocode represents steps without requiring a particular implementation language.
- State assignment, loop bounds and index conventions before tracing.

## Tracing checks one input at a time

- Tracing is a table with one column per variable and one row per step.
- A trace checks the chosen input.
- A claim for all allowed inputs also needs a correctness argument.
- Efficiency: a linear search may inspect all  $n$  items; binary search discards half of an ordered range, needing the sorted-data and bound conventions.

## Worked example: the Euclidean algorithm



- Start:  $a = 10$ ,  $b = 6$ . While  $b$  is nonzero:  $r = a \text{ MOD } b$ ;  $a = b$ ;  $b = r$ .
- Pairs after complete iterations:  $(6,4)$ ,  $(4,2)$ ,  $(2,0)$ . Output: 2.
- Each nonzero remainder is smaller than the previous positive  $b$ , supporting termination.

## Your turn: one more start

### Your turn

Run the same algorithm from  $a = 8$ ,  $b = 6$ . What is the output?

- A. 2, from  $(8,6)$  to  $(6,2)$  to  $(2,0)$
- B. 4, the first remainder
- C. 6, the first  $b$
- D. It never terminates

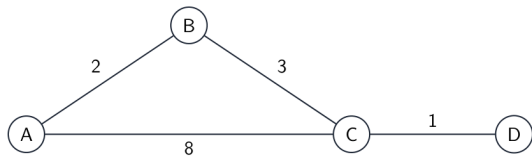
## Graphs, degrees, trees

- A graph is a set of vertices joined by edges, modelling anything with connections.
- For a simple undirected graph, the degree counts incident edges; every edge contributes two to the total degree sum.
- A tree is connected with no cycles; a finite tree with  $n$  vertices has  $n - 1$  edges.
- Some hierarchical models use trees, but actual systems can also contain cross-links or cycles.

## Shortest path, spanning tree

- A shortest path asks for the cheapest route by total weight under the stated constraints — not by the number of edges.
- A minimum spanning tree connects every vertex without cycles, minimising total included edge weight.
- The two tasks can agree on an answer; that does not make them the same task.

## Worked example: weight, not edge count



$$w(A-C-D) = 9; \quad w(A-B-C-D) = 6$$

- Known weights:  $AB = 2$ ,  $BC = 3$ ,  $AC = 8$ ,  $CD = 1$ .
- Path A-C-D has weight 9; path A-B-C-D has weight 6.
- The three-edge path is shorter by weight, despite more edges.

## Your turn: pick the route

### Your turn

Same network. What is the shortest path from A to D, and its weight?

- A. A-B-C-D, weight 6
- B. A-C-D, weight 9
- C. A-B-C-D, weight 5
- D. A-C-D, weight 7

## Check yourself

- 18 in French, 15 in German, 7 both, universe 30: how many are in the union, and how many in neither?
- What is 1011 base 2 in decimal, and what is its hex digit?
- For  $Y = \text{NOT}(A \text{ OR } B)$ , which input row gives  $Y = 1$ ?
- In the AB2/BC3/AC8/CD1 network, which A-to-D route wins, and why?