8.3 Sets and hash tables

Name: _____ Class: _____ Date: _____ **Total: 22 marks**

KEY WORDS union • intersection • difference • set • hash table • hash function • collision • chaining • linear probing • load factor

Objective

Build the skills to answer HL exam questions on **sets and hash tables** – set operations, hash functions, collisions, and the load factor.

You must be able to:

- state the result of **union** , **intersection** and **difference** on two **sets**
- place keys in a **hash table** using a given **hash function**
- resolve a **collision** by **chaining** or **linear probing**
- explain how the **load factor** affects performance

1 Key knowledge and worked examples

Study these first. Each one shows the method for a question type used later.

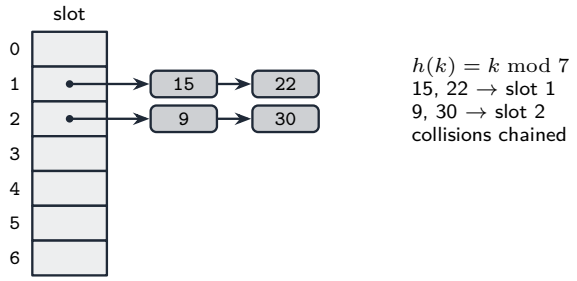
■ **Key knowledge: sets**

A set holds distinct values with no order. For `a = {1, 2, 3, 4}` and `b = {3, 4, 5}`:

Operation	Python	Result
union	`a | b`	`{1, 2, 3, 4, 5}`
intersection	`a & b`	`{3, 4}`
difference	`a - b`	`{1, 2}`
membership	`5 in a`	`False`

- `a.add(2)` changes nothing: 2 is already there.
- Sets and dictionaries are hash tables, so `in` is $O(1)$ on average; on a list it is $O(n)$.

■ **Worked example: hash table with chaining**



Seven slots, $h(k) = k \bmod 7$, keys 15, 22, 9, 30.

1. $15 \bmod 7 = 1$: slot 1.
2. $22 \bmod 7 = 1$: a collision; chaining adds 22 to slot 1's list.
3. $9 \bmod 7 = 2$ and $30 \bmod 7 = 2$: slot 2 holds $9 \to 30$.
4. To find 30: hash to slot 2, then search its short list.

■ **Worked example: linear probing**

Same keys, but each slot holds one key; on a collision, try the next slot, wrapping from 6 to 0.

Key	$k \bmod 7$	Slots tried	Placed in
15	1	1	1
22	1	1, 2	2
9	2	2, 3	3
30	2	2, 3, 4	4

A search for 30 follows the same path and stops at 30, or at an empty slot (not found).

■ **Key knowledge: load factor**

- Load factor $= \dfrac{\text{items stored}}{\text{slots}}$: here $\dfrac{4}{7} \approx 0.57$.
- A higher load factor means more collisions and longer searches; the table is enlarged and every key rehashed.

2 Practice

2.1 $P = \{2, 4, 6, 8\}$ and $Q = \{4, 8, 12\}$. State $P \cup Q$, $P \cap Q$ and $Q - P$. [3]

2.2 Using $h(k) = k \bmod 10$ and 10 slots with chaining, state the slot of each key: 47, 12, 57, 30, 92. Name the slots holding chains. [3]

2.3 Insert 47, 12, 57, 30, 92 into the same table using linear probing instead. State the slot of each key. [3]

2.4 State the load factor of the table in 2.3. [1]

3 Exam-style questions

3.1 Which data structure guarantees that no value is stored twice? [1]

- **A** a list
- **B** a stack
- **C** a set
- **D** a queue

3.2 What is a collision in a hash table? [1]

- **A** two keys hash to the same slot
- **B** the table is full
- **C** a key is not found
- **D** two values are equal

3.3 Why is a hash table usually faster to search than a sorted array? [1]

- **A** it keeps its keys in order
- **B** the hash function gives the slot directly, $O(1)$ on average
- **C** it never has collisions
- **D** it uses less memory

3.4 A school library stores books by 5-digit ID in a hash table of 11 slots, numbered 0 to 10, with $h(k) = k \bmod 11$ and linear probing.

(a) Insert the IDs 10023, 10034, 10045 and 10001 in that order, showing the hash value and slot of each. [4]

(b) State the slots examined when searching for 10045. [1]

(c) Explain why a search for a missing ID can stop at an empty slot. [2]

(d) The library expects 2000 books. Explain why a table of 11 slots is unsuitable, and suggest a size. [2]