

Exercise walk-through

8.3

Sets and hash tables

International Baccalaureate · IB Diploma · Computer Science · HL

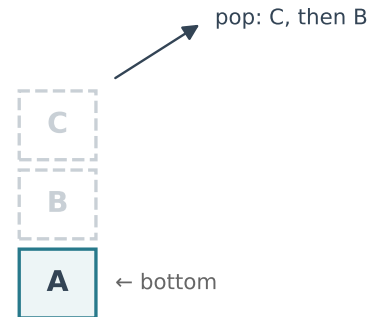
Adt · read the diagram

Queue (FIFO): enqueue A, B, C; dequeue twice; enqueue D



A and B leave from the front, in arrival order.
C, then D, are still waiting.

Stack (LIFO): push A, B, C; pop twice



The last item pushed is the first removed.

- Choose the structure for the required access pattern.
- A queue removes the oldest item first; a stack removes the newest item first

Adt · executable trace

- Trace the state changes and boundary guard.
- Run the boundary case and compare the printed output.

```
from collections import deque
queue = deque()
queue.extend(["A", "B"])
print(queue.popleft())
queue.append("C")
print(queue.popleft())
print(list(queue))
queue.clear()
print(queue.popleft() if queue else "EMPTY")
```

Exercises 2.1·2.2

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]

Exercises 2.1-2.2

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

2.1

- $P \cup Q = \{2, 4, 6, 8, 12\}$
- $P \cap Q = \{4, 8\}$
- $Q - P = \{12\}$

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]

2.2

- $47 \rightarrow 7$, $12 \rightarrow 2$, $57 \rightarrow 7$, $30 \rightarrow 0$, $92 \rightarrow 2$
- chains in slots **2** and **7**

Exercises 2.3-2.4

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]

Exercises 2.3-2.4

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

2.3

- $47 \rightarrow 7, 12 \rightarrow 2, 30 \rightarrow 0$
- 57: 7 taken, slot **8**
- 92: 2 taken, slot **3**

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

2.4

- $\frac{5}{10} = \mathbf{0.5}$

Exercises 3.1-3.2

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

Exercises 3.1-3.2

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

3.1 C

- a set holds distinct values only

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

3.2 A

- two keys, one slot

Exercise 3.3

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

Exercise 3.3

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

3.3 B

- no search through the keys is needed

Exercise 3.4 (a)

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]

Exercise 3.4 (a)

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]

(a)

- $10023 \bmod 11 = 2$: slot **2**
- $10034 \bmod 11 = 2$: taken, slot **3**
- $10045 \bmod 11 = 2$: 2 and 3 taken, slot **4**
- $10001 \bmod 11 = 2$: 2, 3, 4 taken, slot **5**

Exercise 3.4 (b)

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.

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

[1]

Exercise 3.4 (b)

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.

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

(b)

- slots **2, 3, 4**

Exercise 3.4 (c)

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.

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

[2]

Exercise 3.4 (c)

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.

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

(c)

- if the ID had been inserted, it would have gone into the first empty slot on its probe path
- so reaching an empty slot means it is not in the table

Exercise 3.4 (d)

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.

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

Exercise 3.4 (d)

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.

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

(d)

- 11 slots cannot hold 2000 keys with probing; with chaining, chains would be about 180 long, so searches are slow
- a size giving a load factor well below 1, such as 4001 slots (a prime about twice the number of books)