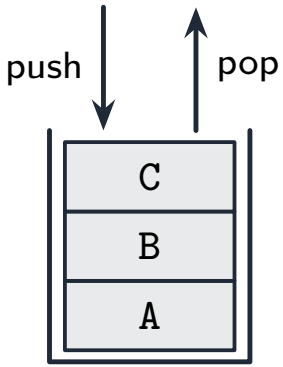
Exercise walk-through

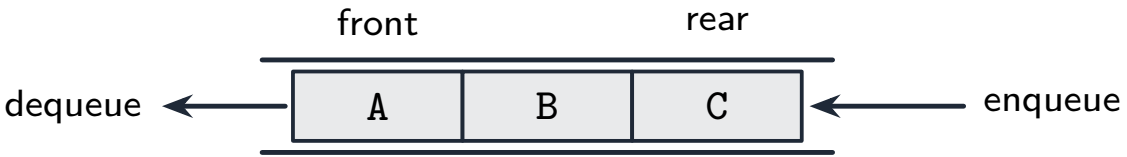
6.3
Searching, sorting and Big O

International Baccalaureate · IB Diploma · Computer Science · HL

Stacks and queues



stack: last in, first out



queue: first in, first out

> Stack: push and pop at the top (undo, back button). Queue: enqueue at the rear, dequeue at the front (print jobs).

Bubble sort and selection sort

After pass	Bubble sort	Selection sort
1	[1, 4, 2, 5]	[1, 5, 4, 2]
2	[1, 2, 4, 5]	[1, 2, 4, 5]
3	no swaps: stop	[1, 2, 4, 5]

Use the evidence

Sorting [5, 1, 4, 2]. **Bubble sort** swaps neighbours; **selection sort** swaps the smallest into place.

Check yourself

Your turn

Exam habits to check before you answer.

A. Does your trace table give a new row each time a variable changes?

B. Did you state that Python indexes from 0?

C. Did you name a stack or queue's order, its two operations and a real use?

D. Did you open files with with and handle bad input with try and except?

E. (AHL) Did you name the base case, the recursive case and how each call moves towards it?

Exercises 2.1·2.3

2.1 List the values compared when a linear search looks for 40 in `[12, 7, 40, 3, 25]`, and the number of comparisons if it looks for 99. [2]

2.2 Show `[5, 2, 8, 1, 4]` after each of the first **two** passes of a bubble sort. [2]

2.3 Show `[5, 2, 8, 1, 4]` after each of the first **two** passes of a selection sort. [2]

Exercises 2.1·2.3

2.1 List the values compared when a linear search looks for 40 in `[12, 7, 40, 3, 25]`, and the number of comparisons if it looks for 99. [2]

2.1

- 12, 7, **40**
- **5** comparisons: every item

2.2 Show `[5, 2, 8, 1, 4]` after each of the first **two** passes of a bubble sort. [2]

2.2

- pass 1: `[2, 5, 1, 4, 8]`
- pass 2: `[2, 1, 4, 5, 8]`

2.3 Show `[5, 2, 8, 1, 4]` after each of the first **two** passes of a selection sort. [2]

2.3

- pass 1: 1 swaps with 5: `[1, 2, 8, 5, 4]`
- pass 2: 2 is already in place: `[1, 2, 8, 5, 4]`

exercises.lol exercises.com.cn

Exercises 2.4·3.1

2.4 A sorted list has 1000 items. State the greatest number of items compared by (i) a linear search; (ii) a binary search. [2]

3.1 Which statement about binary search is true? [1]

- ☐ **A** it works on any list
- **B** it checks every item in turn
- **C** it needs the list to be sorted
- **D** it is $O(n^2)$

Exercises 2.4·3.1

2.4 A sorted list has 1000 items. State the greatest number of items compared by (i) a linear search; (ii) a binary search. [2]

2.4

- (i) **1000**
- (ii) **10**, since $2^{10} = 1024 \geq 1000$

3.1 Which statement about binary search is true? [1]

- ☐ **A** it works on any list
- **B** it checks every item in turn
- **C** it needs the list to be sorted
- **D** it is $O(n^2)$

3.1 C

- it discards the half that cannot hold the target

Exercises 3.2·3.3

3.2 Which is the Big O of a selection sort? [1]

- **A** $O(1)$
- **B** $O(\log n)$
- **C** $O(n)$
- **D** $O(n^2)$

3.3 A bubble sort makes a pass with no swaps. What can be concluded? [1]

- **A** the list is sorted
- **B** the list has duplicates
- **C** one more pass is needed
- **D** the list is in reverse order

Exercises 3.2·3.3

3.2 Which is the Big O of a selection sort? [1]

- ☐ **A** $O(1)$
- ☐ **B** $O(\log n)$
- ☐ **C** $O(n)$
- ☐ **D** $O(n^2)$

3.2 D

- a loop inside a loop

3.3 A bubble sort makes a pass with no swaps. What can be concluded? [1]

- ☐ **A** the list is sorted
- ☐ **B** the list has duplicates
- ☐ **C** one more pass is needed
- ☐ **D** the list is in reverse order

3.3 A

- every neighbouring pair is already in order

Exercise 3.4 (a)

3.4 A library keeps a sorted list of book codes:

```
[104, 117, 131, 148, 152, 166, 179, 183, 190, 205, 218]
```

(a) Trace a binary search for 190, giving `low`, `high`, `mid` and `a[mid]` at each probe. [4]

Exercise 3.4 (a)

3.4 A library keeps a sorted list of book codes:

`[104, 117, 131, 148, 152, 166, 179, 183, 190, 205, 218]`

(a) Trace a binary search for 190, giving `low`, `high`, `mid` and `a[mid]` at each probe. [4]

(a)

`low`	`high`	`mid`	`a[mid]`
0	10	5	166
6	10	8	190

- first probe correct: `mid` 5, 166
- $166 < 190$, so `low` $= 6$
- second probe: `mid` $(6 + 10)//2 = 8$
- 190 found at index **8**

Exercise 3.4 (b)

3.4 A library keeps a sorted list of book codes:

```
[104, 117, 131, 148, 152, 166, 179, 183, 190, 205, 218]
```

(b) State how many items a linear search would compare to find 190. [1]

Exercise 3.4 (b)

3.4 A library keeps a sorted list of book codes:

```
[104, 117, 131, 148, 152, 166, 179, 183, 190, 205, 218]
```

(b) State how many items a linear search would compare to find 190. [1]

(b)

- **9**

Exercise 3.4 (c)

3.4 A library keeps a sorted list of book codes:

```
[104, 117, 131, 148, 152, 166, 179, 183, 190, 205, 218]
```

(c) A new code, 140, is added to the end of the list without sorting it again. Explain why a binary search for 140 now fails. [2]

Exercise 3.4 (c)

3.4 A library keeps a sorted list of book codes:

```
[104, 117, 131, 148, 152, 166, 179, 183, 190, 205, 218]
```

(c) A new code, 140, is added to the end of the list without sorting it again. Explain why a binary search for 140 now fails. [2]

(c)

- the list is no longer sorted
- the first probe finds 166 $>$ 140 and discards the right half, which holds 140, so it returns not found

Exercise 3.4 (d)

3.4 A library keeps a sorted list of book codes:

```
[104, 117, 131, 148, 152, 166, 179, 183, 190, 205, 218]
```

(d) The library has 50 000 books and searches the list many times a day but adds books rarely. Recommend linear or binary search and justify your choice. [3]

Exercise 3.4 (d)

3.4 A library keeps a sorted list of book codes:

```
[104, 117, 131, 148, 152, 166, 179, 183, 190, 205, 218]
```

(d) The library has 50 000 books and searches the list many times a day but adds books rarely. Recommend linear or binary search and justify your choice. [3]

(d)

- **binary** search
- it is $O(\log n)$: about 16 comparisons for 50 000 items, against up to 50 000 for linear
- the list must be kept sorted, which is cheap as books are added rarely