

Exercise walk-through

3.11

Binary Search

AP Computer Science Principles

Lists & binary search

A **list** stores an ordered collection reached by **index** (from 0); **traverse** it with a loop.

Binary search halves the range

Binary search needs a **sorted** list: check the **middle**, then keep only the half that could hold the target. Each step **halves** the range – 1000 items in ~ 10 checks, not 1000.



Search algorithms trade work for speed

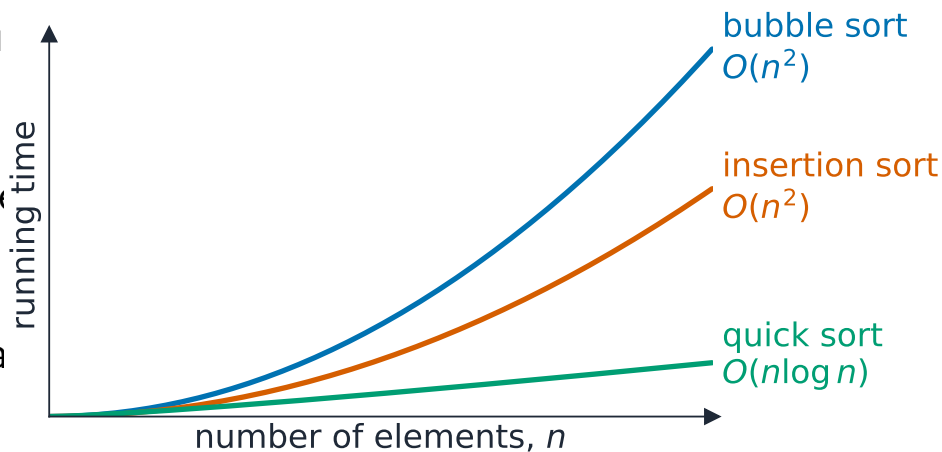
Binary Search

[illegible]

Binary search halves the range at each step (the list must be sorted)

Check yourself

- 1 Binary search on 1,000 items: about many steps?
- 2 What must be true of the list before binary search works?
- 3 Name the three building blocks of an algorithm.
- 4 What makes a problem undecidable?



for large n : $O(n \log n)$ (quick/merge) beats $O(n^2)$ (bubble/insertion)

Answers 1. about 10 ($\log_2 1000$) 2. it must be sorted 3. sequencing, selection, iteration 4. no algorithm can solve it for every input

Exercises 2.1-2.3

2.1 State the requirement a list must meet for binary search to work. [1]

2.2 State what each comparison in a binary search does to the remaining items. [1]

2.3 State one advantage of binary search over linear search. [1]

Exercises 2.1-2.3

2.1 State the requirement a list must meet for binary search to work. [1]

2.1

- it must be sorted
-

2.2 State what each comparison in a binary search does to the remaining items. [1]

2.2

- it halves them (discards one half)
-

2.3 State one advantage of binary search over linear search. [1]

2.3

- it is much faster on large sorted lists (fewer comparisons)

Exercise 3.1

3.1 Binary search requires the list to be

[1]

☐

A sorted

B empty

C reversed

D in random order

Exercise 3.1

3.1 Binary search requires the list to be

[1]

☐

A sorted

B empty

C reversed

D in random order

3.1 A

- sorted.

Exercise 3.2

3.2 Each step of a binary search

[1]

- **A** doubles
- **B** halves
- **C** ignores
- **D** shuffles

the remaining items to search.

Exercise 3.2

3.2 Each step of a binary search

[1]

- **A** doubles
- **B** halves
- **C** ignores
- **D** shuffles

the remaining items to search.

3.2 B

- halves.

Exercise 3.3 (a)

3.3 A sorted list has 16 items.

(a) Name the search method that halves the list at each step.

[1]

Exercise 3.3 (a)

3.3 A sorted list has 16 items.

(a) Name the search method that halves the list at each step.

[1]

(a)

- binary search

Exercise 3.3 (b)

3.3 A sorted list has 16 items.

(b) State roughly how many steps it needs in the worst case ($16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$). [1]

Exercise 3.3 (b)

3.3 A sorted list has 16 items.

(b) State roughly how many steps it needs in the worst case ($16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$). [1]

(b)

- about 4 steps

Exercise 3.3 (c)

3.3 A sorted list has 16 items.

(c) State whether binary search works on an **unsorted** list.

[1]

Exercise 3.3 (c)

3.3 A sorted list has 16 items.

(c) State whether binary search works on an **unsorted** list.

[1]

(c)

- no