

3 Iteration, lists, and efficiency – Part 2

Name: _____ Class: _____ Date: _____

Recall

In Part A you met variables and choices. Now repeating steps and handling many values:

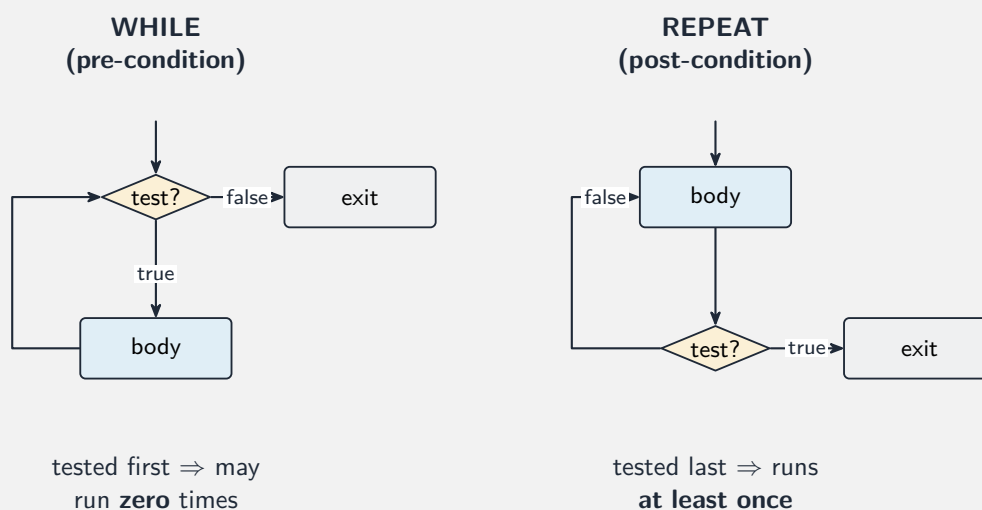
- Doing something 100 times by hand is slow; a loop does it fast.
- A list holds many values under one name.
- Some methods are much faster than others.

Each idea has a worked example before the Practice.

New ideas

■ 1 Iteration

Iteration (a loop) repeats steps. **REPEAT 5 TIMES** runs its block five times; a conditional loop repeats while a test stays true.

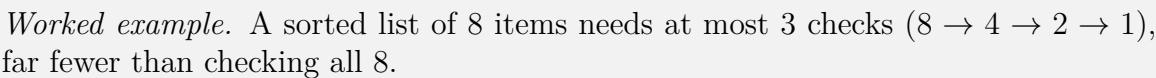


■ 2 Lists

A **list** stores many values in order under one name, each reached by its position (index).

■ 3 Binary search

To find a value in a **sorted** list, **binary search** checks the middle and throws away the half that cannot contain the target, repeating.



An algorithm's **efficiency** is how its work grows with the input size. A **reasonable-time** method grows slowly enough to be practical; an **unreasonable-time** one grows so fast it becomes impossible for large inputs.

Watch out: binary search is only fast because it **halves** the list each step. It only works on a **sorted** list –on an unsorted one you must use the slower linear search.

Try these in order. The methods are all above.

2.5 Explain the difference between a reasonable-time and an unreasonable-time algo-

rithm.

[2]
