# Unit 3 · Algorithms: Searching and Sorting

**WORKSHEET**

Full marks 34

Name _____ Class _____ Roll no. ____ Date _____ Marks _____ / 34

## 1 Key words [4]

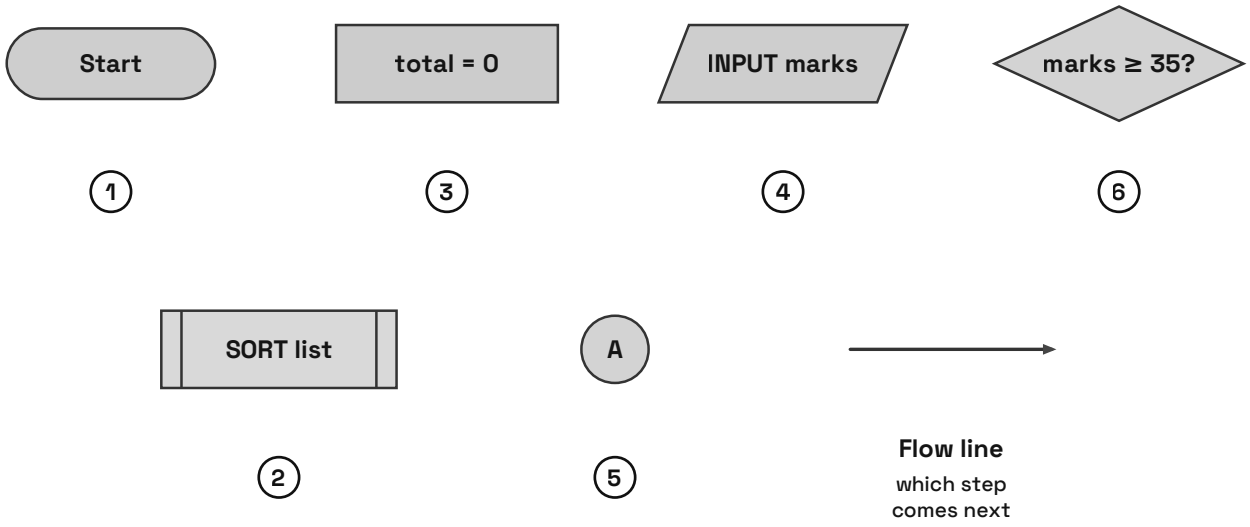
Fill in the blanks with words from the box. Two words in the box are not needed. Each blank carries 0.5 mark.

| **Binary search** | **comparison** | **edge** | **flowchart** | **index** | **IndexError** | **Linear search** | **pass** | **sorted** | **temp** |
|---|---|---|---|---|---|---|---|---|---|

Pseudocode writes an algorithm in short, exact lines of plain words; a _____ draws it as boxes joined by arrows. Each item in a list has a position, also called its _____, counting from 0. _____ looks at the items one by one from the start. _____ looks at the middle of what is left, and works only on a _____ list. Each time an item is compared with the target is one _____. One trip along the whole list in a bubble sort is a _____. A test with the first item, the last item or an empty list is an _____ test.

## 2 Label the diagram [3]

The diagram shows the symbols of a flowchart. Write the name of each symbol in place of 1 to 6. Each carries 0.5 mark.




1. _____ 2. _____

3. _____ 4. _____

5. _____ 6. _____

## 3 Multiple choice [5]

Choose the correct option for each of the following questions.

**i.** `roll = [3, 7, 11, 15, 19]`. What is `roll[len(roll) - 1]`? [1]

**a.** 5 **b.** 15
**c.** 19 **d.** an IndexError

**ii.** Which list can binary search **not** be used on, as it is? [1]

**a.** 2, 5, 9, 14, 20 **b.** Anil, Bina, Chhiring, Deepa
**c.** 14, 3, 27, 8, 19 **d.** 100, 200, 300

**iii.** A sorted list of 63 numbers is searched with binary search. What is the **most** comparisons it can need? [1]

**a.** 63 **b.** 32
**c.** 8 **d.** 6

**iv.** After the first pass of a bubble sort (smallest to biggest), which item is sure to be in its final place? [1]

**a.** the smallest, at the left end **b.** the biggest, at the right end
**c.** the middle one **d.** none of them

**v.** A program finds a student's name from a roll number from 1 to 40. Which is an **edge** test? [1]

**a.** 17 **b.** 40
**c.** 52 **d.** "twelve"

## 4 What does it print? [3]

Write exactly what each program prints, in the box under it.

```
1 marks = [52, 38, 71, 45, 60]
2 marks[1] = 40
3 print(marks[0], marks[4], len(marks))
4 print(marks[len(marks) - 2])
5 print(52 in marks, 38 in marks)
```

OUTPUT

## 5 Trace a binary search [4]

Search the sorted list 4, 9, 13, 18, 25, 31, 36, 42, 50 (positions 0 to 8) for **20**: one row for each look.

| Look | low | high | middle | Number there | So |
|---|---|---|---|---|---|
| 1 | 0 | 8 | 4 | 25 | bigger than 20: high = 3 |
| | | | | | |
| | | | | | |
| | | | | | |

How does the search know that 20 is not in the list?

## 6 Sort by hand [3]

Sort the list, then answer.

Sort 29, 14, 35, 8 from smallest to biggest with bubble sort. Show the list after each pass, and count the comparisons and swaps.

## 7 Find and fix the bug [3]

Read the program, then answer.

This linear search stops with `IndexError: list index out of range`.

```
1  names = ["Asha", "Binod", "Chhiring", "Dipa", "Esha"]
2  target = "Kopila"
3  i = 0
4  found = False
5  while i <= len(names) and not found:
6      if names[i] == target:
7          found = True
8      else:
9          i = i + 1
10 print(found)
```

i. Which line has the mistake, and why does it stop the program?
ii. Write that line correctly.
iii. Why would a search for Dipa not show the bug?

## 8 Write a program [4]

Write the program in Python on the lines. Any program that gives the right result gets the marks.

`names = ["Sita", "Bikash", "Gita", "Hari", "Maya", "Nabin"]`. Write a program that asks for a name and finds it with a linear search, without using `in`. It prints `Hari is at position 3` for Hari, and `Asha is not in the list.` for Asha.

## 9 Choosing tests [3]

Write short answers to the following question.

A program finds a student's name from a roll number from 1 to 40. Give one **ordinary** test, one **edge** test and one **wrong** test for it, each with the result it should give.

## 10 Think [2]

Think, then answer.

If binary search is so fast, why does anyone use linear search?

## 11 Activity: Sort like a computer (not marked)

Work in a group of three, with 8 slips of paper, each with a different number from 1 to 99, shuffled and face down in a row. The computer may turn over only **two** slips at a time; one student tells it which two, following the sort's plan; one counts. Sort with each plan, then shuffle and do it again.

| Sort | Comparisons, first shuffle | Comparisons, second shuffle | Which needed fewer? |
|---|---|---|---|
| Bubble sort | | | |
| Insertion sort | | | |