Unit 2 · Graphs: Maps, Routes and Networks

WORKSHEET

Full marks 33

Name _____ Class _____ Roll no. ____ Date _____ Marks _____ / 33

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.

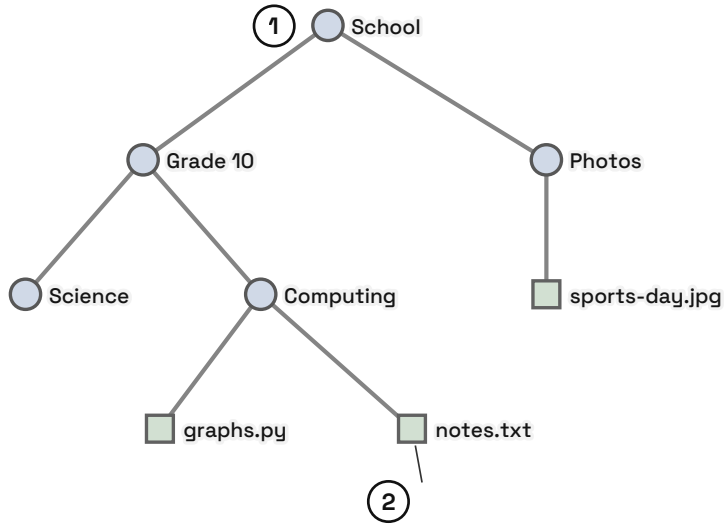
degree	directed	edge	leaf	node	queue	root	stack	tree	weighted

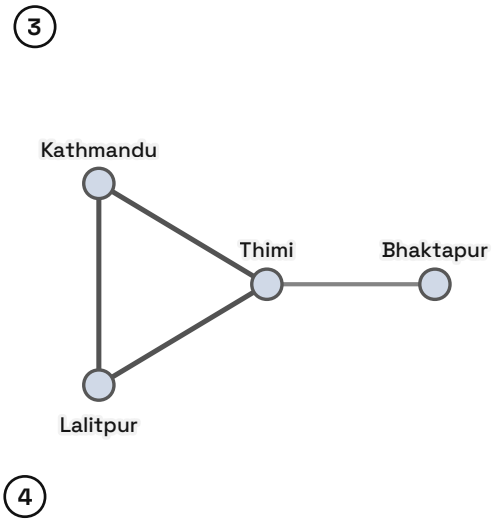
A graph is a set of things, each a _____, and the connections between them, each an _____. The number of edges at a node is its _____. A graph whose edges go only one way, drawn as arrows, is _____; a graph with a length, a time or a cost on each edge is _____. Breadth-first search keeps the places still to visit in a _____: added at the back, taken from the front. A joined-up graph with no loops is a _____, and a node in it with no children is a _____.

2 Label the diagram [2]

The diagram shows two graphs. Write the right word or words in place of 1 to 4. Each carries 0.5 mark.

A tree: a folder tree





1. _____ 2. _____

3. _____ 4. _____

3 Multiple choice [5]

Choose the correct option for each of the following questions.

i. Which of these is best shown as a **directed** graph? [1]

a. towns joined by two-way roads **b.** students who sit next to each other

c. who follows whom on a social media app **d.** countries that share a border

ii. A program stores a graph of 5,000 villages and the trails between them. Most villages have two to four trails. Its most common job is "list this village's neighbors". Which structure fits best? [1]

a. a table with a row and a column for every village

b. a list of every trail

c. a dictionary with each village's list of neighbors

d. one long string of all the village names

iii. In breadth-first search, where does a newly seen place go, and which place is visited next? [1]

a. on the front of the list; the next is taken from the back

b. on the back of the list; the next is taken from the front

c. anywhere in the list; the next is the nearest on the map

d. on the back of the list; the next is taken from the back

iv. A breadth-first search from one town reaches 200 towns and reads 600 lists of neighbors in all. A graph twice as big, with 400 towns and twice as many roads, is searched in the same way. About how much work will it do? [1]

a. the same amount

b. about twice as much

c. about four times as much

d. about 200 times as much

v. A tree has 25 nodes. How many edges does it have? [1]

a. 25

b. 26

c. 24

d. 50

4 What does it print? [3]

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

The paths between the buildings of a school, as a dictionary of neighbors:

```
paths = {
    "Gate": ["Office", "Library"],
    "Office": ["Gate", "Lab"],
    "Library": ["Gate", "Lab", "Canteen"],
    "Lab": ["Office", "Library"],
    "Canteen": ["Library"],
}

ends = 0
for place in paths:
    ends = ends + len(paths[place])
    if len(paths[place]) == 1:
        print(place, "is a dead end")
print(ends // 2, "paths")
print("Lab" in paths["Gate"], len(paths["Library"]))
```

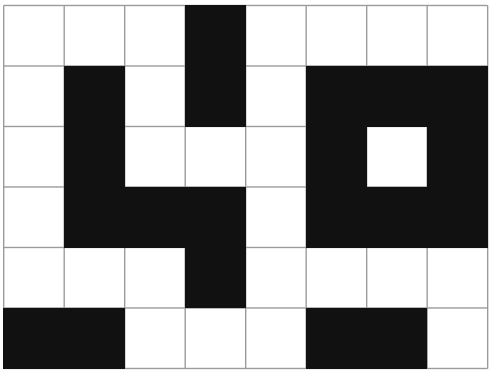
```
OUTPUT
```

5 Paint it, ring by ring [4]

Read, then answer.

A paint bucket fills the white squares of this picture by breadth-first search, starting from the top-left square. A square's neighbors are the white squares above, below, left and right of it (never across a corner); black squares are outlines.

i. Write in every white square that gets painted its number of steps from the start: 0 in the top-left square, 1 in its neighbors, and so on.



Start: the top-left square, 0.

ii. Which square is painted last, and after how many steps? How many squares are painted in all?

iii. One white square is never painted. Which, and why?

6 Trace a search [4]

Carry out breadth-first search by hand on this graph of friends, starting from Hari. Write a row for each visit. The first row is filled in.

```
friends = {
    "Asha": ["Bimal", "Dawa", "Gita"],
    "Bimal": ["Asha", "Hari", "Nima"],
    "Dawa": ["Asha", "Gita", "Nima"],
    "Gita": ["Asha", "Dawa", "Nima"],
    "Hari": ["Bimal"],
    "Nima": ["Bimal", "Dawa", "Gita"],
}
```

Visit	Steps	New places (steps)	List still to visit, front first
(start)		Hari (0)	Hari

7 Find and fix the bug [3]

Read the program, then answer.

Seven junctions of a ring road, each joined to the next. The program should print how many roads each junction is from Kalanki, but it prints 5 for Satdobato, though Kalanki → Balkhu → Satdobato is only 2.

```
ring = {
    "Kalanki": ["Balkhu", "Balaju"],
    "Balkhu": ["Kalanki", "Satdobato"],
    "Satdobato": ["Balkhu", "Koteshwor"],
    "Koteshwor": ["Satdobato", "Chabahil"],
    "Chabahil": ["Koteshwor", "Gongabu"],
    "Gongabu": ["Chabahil", "Balaju"],
    "Balaju": ["Gongabu", "Kalanki"],
}

def breadth_first(graph, start):
    steps = {start: 0}
    to_visit = [start]
    while len(to_visit) > 0:
        place = to_visit.pop()
        for nxt in graph[place]:
            if nxt not in steps:
                steps[nxt] = steps[place] + 1
                to_visit.append(nxt)
    return steps

print(breadth_first(ring, "Kalanki"))
```

i. Which line is wrong? Write it correctly.

ii. Why does the mistake give Satdobato 5 steps?

8 Write a program [4]

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

Write a function `farthest(graph, start)` that uses breadth-first search to give back the place that is the most steps from **start**, and its number of steps. For the valley's `roads`, `farthest(roads, "Kathmandu")` should give `('Dhulikhel', 4)`.

9 Short answer [2]

Write a short answer to the following question.

A map app gives a route of 5 roads from Bhaktapur to Thamel, though there is a route of only 3 roads. Explain how this can be the right answer, and why breadth-first search alone couldn't find it.

10 Think [2]

Think, then answer.

Look at the school's **paths** in **What does it print?** Could a guard walk along every path exactly once, without walking any path twice? If so, where must the walk start and end? Give a walk.

11 Activity: Search the valley (not marked)

At a computer, open the unit's page (the QR code) and find **Search the valley.** Tap each town to make it the goal, press **Finish**, and follow the arrows back to Kathmandu.

Goal	Steps from Kathmandu	The route, from Kathmandu
Dhulikhel		
Banepa		
Godawari		
Pharping		
Sankhu		