



U9.2 A grid is a graph

Name _____

Class _____

Date _____

What this lesson is about

Cells are nodes, neighbours are edges, and breadth first search is the shortest path when every step costs the same.

- The grid used from here on
- Choosing the cell size
- Breadth first search
- Four neighbours or eight

Questions

6 marks in all

1. What does this program print?

[1 mark]

```
rows = ["S..#",
        ".#.#",
        "...G"]
H, W = len(rows), len(rows[0])
start, goal = (0, 0), (2, 3)
came = {start: None}
queue = [start]
expanded = 0
while queue:
    cur = queue.pop(0)
    expanded += 1
    if cur == goal:
        break
    r, c = cur
    for nxt in ((r + 1, c), (r - 1, c), (r, c + 1), (r, c - 1)):
        nr, nc = nxt
        if 0 <= nr < H and 0 <= nc < W and rows[nr][nc] not in "#" and nxt not in came:
            came[nxt] = cur
            queue.append(nxt)
steps, cur = 0, goal
while came[cur] is not None:
    cur = came[cur]
    steps += 1
print(steps, expanded)
```

2. Breadth first search is run on an eight-connected grid. What is wrong with the path it returns?

[1 mark]

- ☐ **A** It counts a diagonal step as one, the same as a straight step, so the path has fewest steps but need not be shortest
- ☐ **B** It may not find a path even when one exists
- ☐ **C** It visits cells more than once and may loop for ever
- ☐ **D** It returns a path that passes between diagonal cells through a wall

3. On the U9 grid (5 cm cells), a four-connected path runs from cell (6, 6) to cell (34, 34) with no detours. [1 mark]
How long is it, in cm?
-
4. In the breadth first code, the dictionary came does two jobs. What are they? [1 mark]
- ☐ A It is the visited set, and it records which cell each cell was reached from so the path can be walked back
 - ☐ B It stores the queue, and it stores the cost of each cell
 - ☐ C It stores the free cells, and it stores the walls
 - ☐ D It records the goal, and it counts the steps
5. On the U9 grid, what is the x coordinate, in cm, of the centre of cell (13, 20)? [1 mark]
-
6. The cell size is made much larger and breadth first now reports no route across the mat. Why? [1 mark]
- ☐ A No cell centre lands in the passage between the walls, so the passage has disappeared from the graph
 - ☐ B Breadth first is not complete on coarse grids
 - ☐ C The queue overflows with large cells
 - ☐ D Larger cells make every edge cost different

The task: breadth first across the mat

Build the grid, run breadth first from cell (6, 6) to cell (34, 34) with four neighbours, and print `steps:`, how many moves long the path is, and `expanded:`, how many cells came off the front of the queue before the goal did.

```
from bugbot import *
connect()

CELL = 5.0
N = 40
INFLATE = 8.0
WALLS = [(70.0, 0.0, 8.0, 115.0), (125.0, 85.0, 8.0, 115.0)]
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/u9-2-a-grid-is-a-graph/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Count how many cells breadth first expanded and compare it with the number of free cells. What does the ratio tell you about how much of the mat it searched for nothing?
2. Run it with the goal set to a cell inside a wall. What happens, and how long does it take to happen?
3. Run it with eight neighbours and measure the true length of the path in centimetres. Is it shorter or longer than the four-connected one?