Exercise walk-through

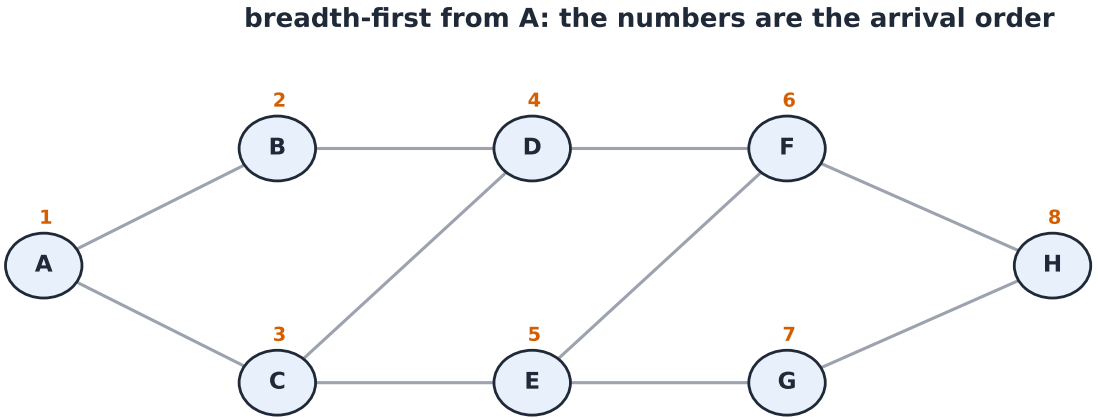
5.5
Graphs and traversal

University Computer Science

Graphs and traversal: two probes, one graph

1. **Storage** 8 nodes, 10 undirected edges: the **adjacency list** holds 20 entries, the matrix 64 cells - the list wins sparse, the matrix dense.
2. **BFS** From A the layers read A, then B C, then D E, then F G, then H - the layer is the shortest hop count.
3. **DFS** From A the deep walk reads A, B, D, C, E, F, H, G - one path with backtracks.

Graph with traversal order



layers: A | B C | D E | F G | H — the layer is the shortest hop count from A

The course graph: breadth-first from A visits in the numbered order A, B, C, D, E, F, G, H - layer by layer.

Exercises 2.1·2.3

2.1 An undirected graph has 12 edges. Give its adjacency-list entry count and its matrix size for 9 nodes. [3]

2.2 What does a node's BFS layer equal, and under what weight assumption? [3]

2.3 State the one marking rule both traversals share, and what breaks without it. [3]

Exercises 2.1·2.3

2.1 An undirected graph has 12 edges. Give its adjacency-list entry count and its matrix size for 9 nodes. [3]

2.1

- List: 24 entries (each undirected edge twice) ✓.
- Matrix: $9 \times 9 = 81$ cells ✓.

2.2 What does a node's BFS layer equal, and under what weight assumption? [3]

2.2

- The shortest hop count from the start ✓.
- Unweighted edges only – weights need Dijkstra ✓.

2.3 State the one marking rule both traversals share, and what breaks without it. [3]

2.3

- Mark each node visited once and never re-enqueue it ✓.
- Without it, any cycle loops forever ✓.

Exercise 3.1

3.1 A graph has vertices P, Q, R, S, T and undirected edges PQ, PR, QS, RS, ST.

Write its adjacency list. Run BFS from P with neighbours visited alphabetically: give the visit order and each node's layer. Run DFS from P the same way and give the visit order. State the cost of each traversal on your list, and on the adjacency matrix of this graph. [8]

Exercise 3.1

3.1 A graph has vertices P, Q, R, S, T and undirected edges PQ, PR, QS, RS, ST.

Write its adjacency list. Run BFS from P with neighbours visited alphabetically: give the visit order and each node's layer. Run DFS from P the same way and give the visit order. State the cost of each traversal on your list, and on the adjacency matrix of this graph. [8]

3.1

- List: P:Q, R; Q:P, S; R:P, S; S:Q, R, T; T:S ✓.
- BFS: P, Q, R, S, T with layers $P \,|\, Q, R \,|\, S \,|\, T$ ✓.
- DFS: $P \to Q \to S \to R$ (backtrack) $\to T$: order P, Q, S, R, T ✓.
- On the list: $O(V + E) = O(5 + 5)$ ✓.
- On the matrix: $O(V^2) = 25$ cells scanned ✓.

Exercise 3.2

3.2 Choose the storage for each scenario and justify: a social network with 10^6 users and average degree 50; a timetable lookup asking only whether station X connects directly to station Y among 300 stations, half the pairs connected.

Compute the list-entry count and the matrix-cell count for the network case, and state which store wins each scenario by comparing the counts. [7]

Exercise 3.2

3.2 Choose the storage for each scenario and justify: a social network with 10^6 users and average degree 50; a timetable lookup asking only whether station X connects directly to station Y among 300 stations, half the pairs connected.

Compute the list-entry count and the matrix-cell count for the network case, and state which store wins each scenario by comparing the counts. [7]

3.2

- Network: $V = 10^6$, $E = 10^6 \times 50/2 = 2.5 \times 10^7$ ✓.
- List entries: 5×10^7; matrix cells: 10^{12} – list wins by four orders of magnitude ✓.
- Timetable: 300 stations, about half of $300^2 = 90\,000$ cells set – dense ✓.
- The question is one cell lookup: the matrix answers in $O(1)$ – matrix wins ✓.