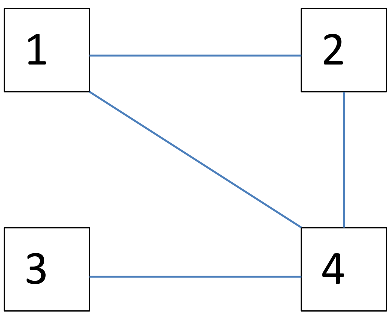
Graphs

Graph G = (V,E) with n=|V|, m=|E|

- Adjacency matrix data structure
- Adjacency lists data structure

Example: Unweighted Undirected Graph



Adjacency matrix
M[1…n][1…n] of booleans
M[j][k] = 1 iff (j,k) is edge
M[j][k] = M[k][j]

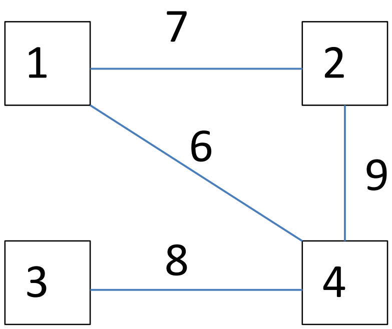
M	1	2	3	4
1	0	1	0	1
2	1	0	0	1
3	0	0	0	1
4	1	1	1	0

Adjacency lists
L[1…n] of lists of vertices
Each L[k] has neighbors of k
(usually in ascending order)

L

1 → 2 → 4
2 → 1 → 4
3 → 4
4 → 1 → 2 → 3

Example: Weighted Undirected Graph



Weighted adjacency matrix
M[1…n][1…n] of ints/floats
M[j][k] = weight(j, k)
M[j][k] = M[k][j]

Weighted adjacency lists
L[1…n] of lists of pairs
[neighbor, edge weight]

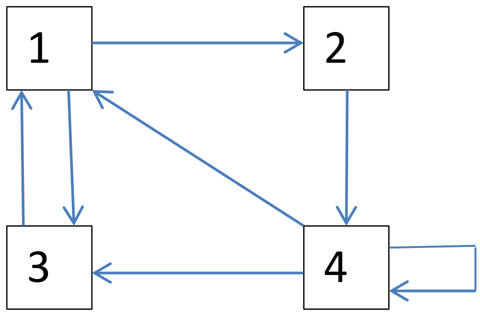
M	1	2	3	4
1	0	7	∞	6
2	7	0	∞	9
3	∞	∞	0	8
4	6	9	8	0

L

1 → [2,7] → [4,6]
2 → [1,7] → [4,9]
3 → [4,8]
4 → [1,6] → [2,9] → [3,8]

Typically use 0 or ∞ for
non-existent edges

Example: Unweighted Directed Graph



Adjacency matrix
M[1…n][1…n] of booleans
M[j][k] = 1 iff (j,k) is edge
M[j][k] can be $\neq$ M[k][j]

M	1	2	3	4
1	0	1	1	0
2	0	0	0	1
3	1	0	0	0
4	1	0	1	1

Adjacency lists
Out[1…n], In[1…n]
Out[k] has outgoing neighbors
In[k] has incoming neighbors

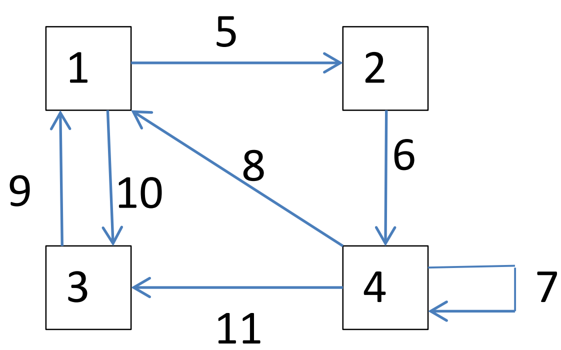
Out

1 $\to$ 2 $\to$ 3
2 $\to$ 4
3 $\to$ 1
4 $\to$ 1 $\to$ 3 $\to$ 4

In

1 $\to$ 3 $\to$ 4
2 $\to$ 1
3 $\to$ 1 $\to$ 4
4 $\to$ 2 $\to$ 4

Example: Weighted Directed Graph



Weighted adjacency matrix
M[1…n][1…n] of ints/floats
M[j][k] = weight(j, k)
M[j][k] can be $\neq$ M[k][j]

	1	2	3	4
1	0	5	10	∞
2	∞	0	∞	6
3	9	∞	0	∞
4	8	∞	11	7

Weighted adjacency lists
Out[1…n], In[1…n]
Out[k] and In[k] are lists of pairs
[neighbor, edge weight]

Out

1 →[2,5]→[3,10]
2 →[4,6]
3 →[1,9]
4 →[1,8]→[3,11]→[4,7]

In

1 → [3,9] → [4,8]
2 → [1,5]
3 → [1,10] → [4,11]
4 → [2,6] → [4,7]

Analysis:

Each kind of adjacency matrix:

$\theta(n^2)$ space/memory used

Each kind of adjacency lists:

$\theta(n+m)$ space/memory used

Recall m = number of edges

Upcoming we will look at several algorithms on graphs. We'll analyze the running time of each algorithm using the adjacency matrix and/or adjacency lists data structures.