Dynamic Programming: Transitive Closure

(Chapter 8)

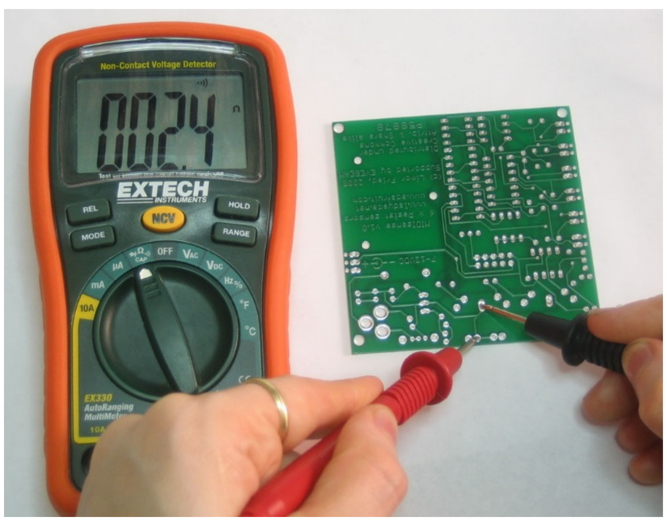
Transitive Closure

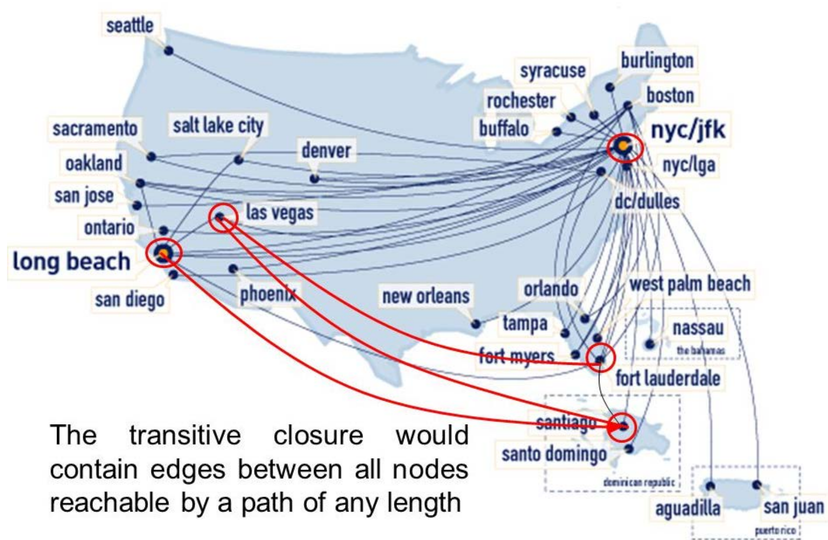
- What nodes are reachable from other nodes?
- Problem:
  - given a directed unweighted graph G with n vertices, find all paths that exist from vertices v_i to v_j, for all $1 \leq (i, j) \leq n$

- Note: this problem is always solved with an adjacency matrix graph representation

Transitive Closure

- Applications:
  - Testing digital circuits, reachability testing

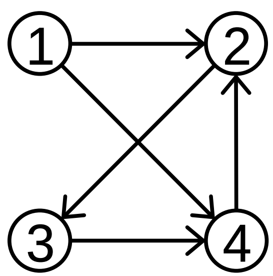




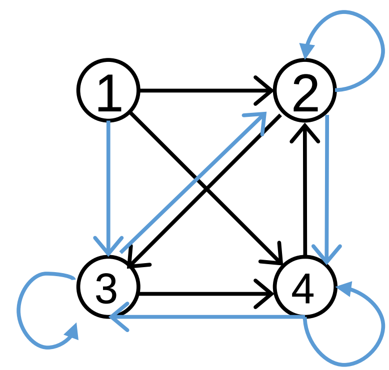
The transitive closure would contain edges between all nodes reachable by a path of any length

Transitive Closure

- Idea of algorithm:
  - Create a new graph where every ***edge*** represents a ***path*** in the original



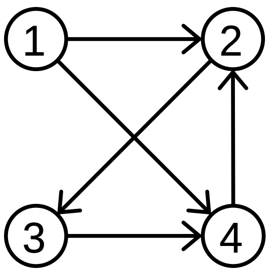
	1	2	3	4
1	0	1	0	1
2	0	0	1	0
3	0	0	0	1
4	0	1	0	0



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

Transitive Closure example

- Consider the graph below, and its corresponding adjacency matrix ...



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

- We call this initial matrix R^0.
  - For convenience here we are using a 1-based array: A[1..n][1..n]

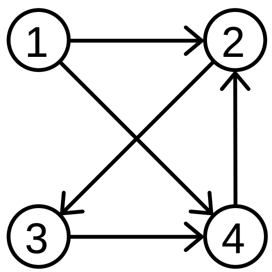
Transitive Closure

Step 1:

- select row 1 and column 1
- for all i,j

  if (i,1) = 1 and (1,j) = 1 then set (i,j) ← 1

In this case there are no changes.



j

i

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

At the end of this step this matrix is known as R^1.

Transitive Closure

Step 2:

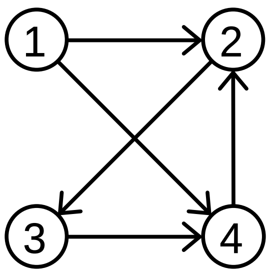
- select row 2 and column 2
- for all i,j

    if (i,2) = 1 and (2,j) = 1 then set (i,j) ← 1

Notice:

(1,2) == (2,3) == 1 → set (1,3) ← 1

(4,2) == (2,3) == 1 → set (4,3) ← 1



			j		
		1	2	3	4
	1	0	1	**1**	1
i	2	0	0	1	0
	3	0	0	0	1
	4	0	1	**1**	0

At the end of this step this matrix is known as R^2.

Transitive Closure

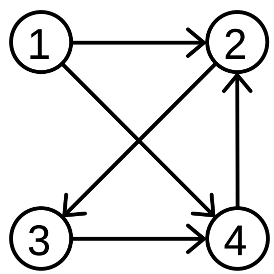
Step 3:

- select row 3 and column 3
- for all i,j

    if (i,3) = 1 and (3,j) = 1 then set (i,j) ← 1

Notice:

(1,3) == (3,4) == 1 → set (1,4) ← 1
(2,3) == (3,4) == 1 → set (2,4) ← 1
(4,3) == (3,4) == 1 → set (4,4) ← 1



i \ j	1	2	3	4
1	0	1	1	**1**
2	0	0	1	**1**
3	0	0	0	1
4	0	1	1	**1**

At the end of this step this matrix is known as R^3.

Transitive Closure

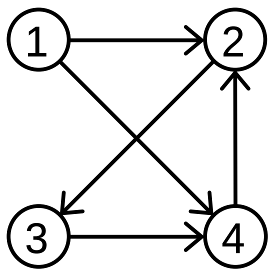
Step 4:

- select row 4 and column 4
- for all i,j

  if (i,4) = 1 and (4,j) = 1 then set (i,j) ← 1

Notice:

(2,4) == (4,2) == 1 → set (2,2) ← 1
(3,4) == (4,2) == 1 → set (3,2) ← 1
(3,4) == (4,3) == 1 → set (3,3) ← 1



j

		1	2	3	4
	1	0	1	1	1
i	2	0	**1**	1	1
	3	0	**1**	**1**	1
	4	0	1	1	1

At the end of this step this matrix is known as R^4. It is the "Transitive Closure on G". The existence of a one in cell (i,j) tells us that there exists a path from i to j in G.

Warshall's algorithm

- Maybe the best thing about this algorithm is its simplicity

```
Warshall(G[1..n, 1..n])
    for k ← 1 to n {
        for i ← 1 to n {
            for j ← 1 to n {
                if ( G[i,k] == G[k,j] == 1 ) {
                    set G[i,j] ← 1
                }
            }
        }
    }
```

Efficiency: ?

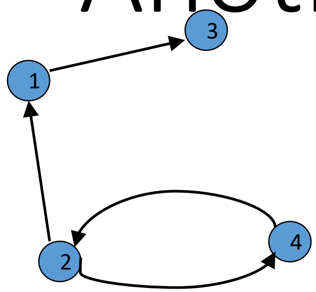
Why is this Dynamic Prog?

- On the *k*-th iteration:
  - The algorithm determines for every pair of vertices *i, j* if a path exists from *i* and *j* with just vertices 1,…,*k* allowed as intermediate

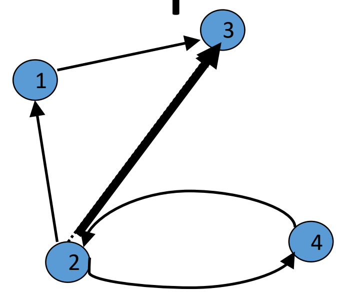
- So: It finds the paths from simpler subproblems

- Also produces the result bottom-up from a matrix recording as you go

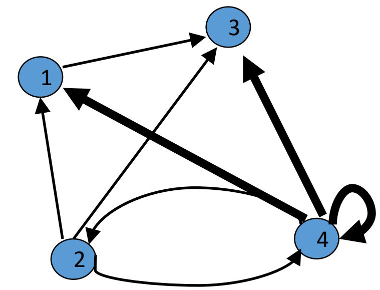
Another Example



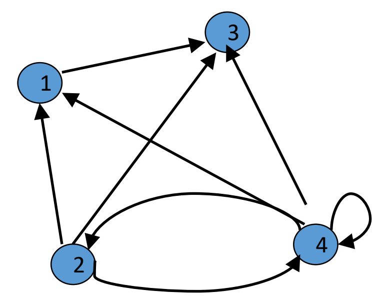
$$R^{(0)} = \begin{matrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$



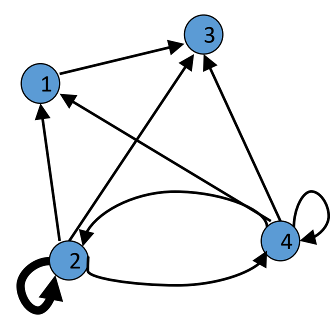
$$R^{(1)} = \begin{matrix} 0 & 0 & 1 & 0 \\ 1 & 0 & \mathbf{1} & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$



$$R^{(2)} = \begin{matrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ \mathbf{1} & 1 & \mathbf{1} & \mathbf{1} \end{matrix}$$



$$R^{(3)} = \begin{matrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{matrix}$$



$$R^{(4)} = \begin{matrix} 0 & 0 & 1 & 0 \\ 1 & \mathbf{1} & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{matrix}$$