Matrix Multiplication
and Graph Search
Standard $O(n^3)$ matrix multiplication

input: $n \times n$ matrices $A$ and $B$ (of int)
output: product $C = A \times B$

for $i=1$ to $n$
  for $j=1$ to $n$
    $C[i,j] = 0$
    for $k=1$ to $n$
      $C[i,j] = C[i,j] + A[i,k] \times B[k,j]$
Matrix multiplication over \{0,1\}

input: \( n \times n \) matrices \( A \) and \( B \) (of boolean)
output: product \( C = A \times B \)

\[
\begin{align*}
\text{for } i = 1 \text{ to } n \\
&\quad \text{for } j = 1 \text{ to } n \\
&\quad \quad C[i,j] = \text{false} \\
&\quad \quad \text{for } k = 1 \text{ to } n \\
&\quad \quad \quad C[i,j] = C[i,j] \lor (A[i,k] \land B[k,j])
\end{align*}
\]

+ becomes OR (\( \lor \)) and * becomes AND (\( \land \))
Transitive closure

• \( M \) is the adjacency matrix

• \( M^2 \) (using boolean matrix mult) tells us about paths of length 2

• ... and \( M^k \) about paths of length \( k \)

• the only \( k \) that matter are \( 0 \leq k < V \)

• \( M^* = M^0 + M^1 + M^2 + ... + M^{V-1} \)

• \( M^* = (I+M)^V \)
Shortest paths (future)

for $i=1$ to $n$
  for $j=1$ to $n$

  $W^{\leq 2}[i,j] = [\text{if } i=j \text{ then } 0 \text{ else } \infty]$

  for $k=1$ to $n$
    $W^{\leq 2}[i,j] = \text{MIN}(W^{\leq 2}[i,j], W[i,k]+W[k,j])$
Breadth-First Search (from page 595 CLRS)

BFS(G,s)
1   for each vertex u in V-{s}
2     u.color = WHITE
3     u.dist = infinity
4     u.prev = nil
5   s.color = GRAY
6   s.dist = 0
7   s.prev = nil
8   Q = empty
9   ENQUEUE(Q,s)
10  while Q not empty
11    u = DEQUEUE(Q)
12    for each v in ADJ(u)  -- adjacency list of u
13       if v.color = WHITE
14         v.color = GRAY
15         v.dist = u.dist + 1
16         v.prev = u
17         ENQUEUE(Q,v)
18    u.color = BLACK
Depth-First Search (page 604 CLRS)

DFS(G)
1  for each vertex u in V
2     u.color = WHITE
3     u.prev = nil
4  time = 0
5  for each vertex u in V
6     if u.color = WHITE
7        DFS-Visit(G,u)

DFS-Visit(G,u)
1  time = time + 1
2  u.disc = time
3  u.color = GRAY
4  for each v in adjacency list of u
5      if v.color = WHITE
6         v.prev = u
7         DFS-Visit(G,v)
8  u.color = BLACK
9  time = time + 1
10 u.finish = time

white - not seen yet
gray - in process
black - done