Matrix Multiplication and Graph Search

Standard O(n^3) matrix multiplication

```
input: n×n matrices A and B (of int)
output: product C = A*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]*B[k,j]
```

Matrix multiplication over {0,1}

```
input: n×n matrices A and B (of boolean)
output: product C = A*B

for i=1 to n
    for j=1 to n
        C[i,j] = false
        for k=1 to n
            C[i,j] = C[i,j]∨(A[i,k]∧B[k,j])
```

+ becomes OR (∨) and * becomes AND (∧)

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 $0 \le k < V$
- $M^* = M^0 + M^1 + M^2 + \ldots + M^{V-1}$
- $M^* = (I+M)^V$

Shortest paths (future)

```
for i=1 to n
    for j=1 to n

        W≤2[i,j] = [if i=j then 0 else ∞]

        for k=1 to n
          W≤2[i,j] = MIN(W≤2[i,j], W[i,k]+W[k,j])
```

Breadth-First Search (from page 595 CLRS)

```
BFS(G,s)
for each vertex u in V-{s}
   u.color = WHITE
   u.dist = infinity
   u.prev = nil
s.color = GRAY
s.dist = 0
s.prev = nil
Q = empty
ENQUEUE(Q,s)
while Q not empty
   u = DEQUEUE(Q)
   for each v in ADJ(u)  -- adjacency list of u
      if v.color = WHITE
          v.color = GRAY
          v.dist = u.dist + 1
          v.prev = u
          ENQUEUE(Q,v)
   u.color = BLACK
```

Depth-First Search (page 604 CLRS)

```
DFS(G)
for each vertex u in V
   u.color = WHITE
   u.prev = nil
time = 0
for each vertex u in V
   if u.color = WHITE
      DFS-Visit(G,u)
```

```
DFS-VISIT(G,u)
time = time + 1
u.disc = time
u.color = GRAY
for each v in adjacency list of u
   if v.color = WHITE
      v.prev = u
      DFS-Visit(G,v)
u.color = BLACK
time = time +1
u.finish = time
```

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