PSPACE

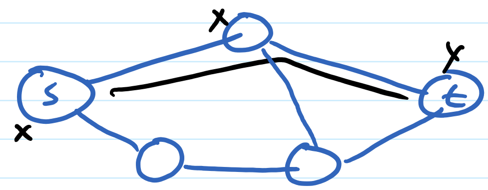
```
DFS_VISIT(curr, parent, t)
{
  // mark current vertex as processing and do bookkeeping
  color[curr] = GRAY;
  pi[curr] = parent;

  // iterate over outgoing edges
  for each vertex v s.t. there is an edge (curr, v)
      if (color[v] == WHITE)
          // found an edge to a new vertex -- explore it
          DFS_VISIT(v, curr, t);

  // mark current vertex finished
  s->color[curr] = BLACK;
}
```

if curr = t then
if π defines a HP return YES

WHITE




```
INDEPENDENT_SET(G, k)
{
  n = number of vertices in G
  best = empty

  for i = 0 to 2^n - 1
    // interpret i as a set
    S = empty
    for j = 0 to n-1
      if i % 2^j != 0
        S = S union {j}

    if S is an independent set in G and len(S) > len(best)
      best = S

  return len(best) >= k
}
```

n=5

00000 0

01001 → {0, 3}

2^3 2^0

11111 $31 = 2^5 - 1$

poly space so IS ∈ PSPACE

IS ∈ PSPACE

HC(G)

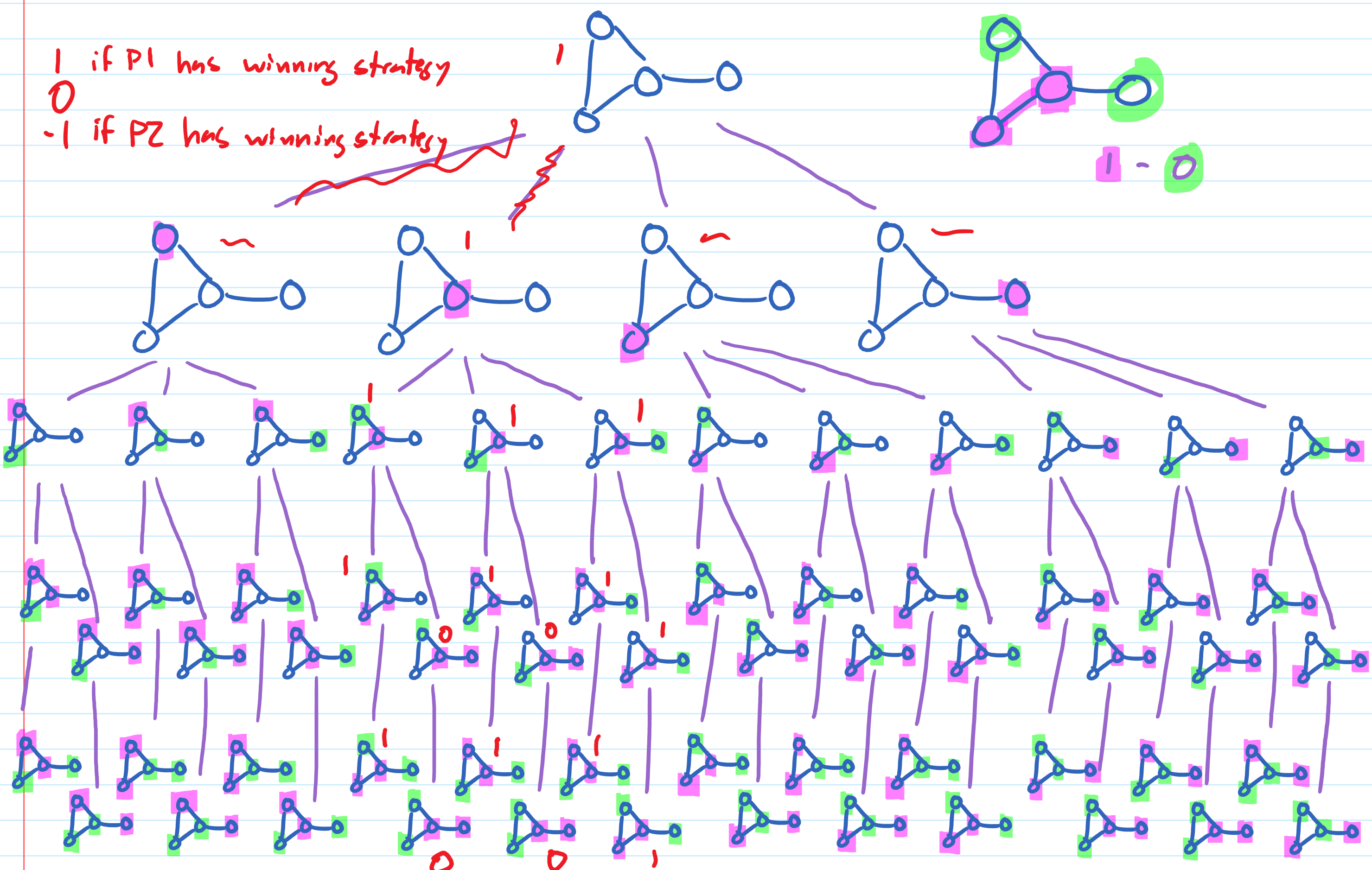
for i=0 to n!-1
interpret i as permutation of vertices P
if p + edge to complete cycle is a cycle in G
return YES
return NO

HC ∈ PSPACE

Finite Combinatorial Games

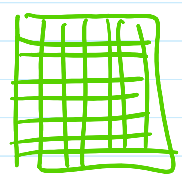
Edge Covering Game: 2 players take turns coloring an uncolored vertex in an undirected graph their color. Winner is player with most edges having their color at both ends.

1 if P1 has winning strategy
0
-1 if P2 has winning strategy



MINIMAX(p)

if p is end of game
  return value according to rules
else
  let S = set of positions reachable in one turn from p
  if p is P1's turn
    return $\max_{p' \in S}$ MINIMAX(p')
  else
    return $\min_{p' \in S}$ MINIMAX(p')



$\leq 13^{64}$ arrangement

polynomial space per call for most games

length of game polynomial in size of board
↓
polynomial recursion depth
↓
poly·poly total space → poly total space

putting pieces on board / no moving or removing
Othello

putting pieces on board / no moving or removing

Othello
Tic-Tac-Toe
Gomoku
Connect Four

$\in$ PSPACE

poly · poly total space → poly total space

↓

same $\in$ PSPACE

$P \subseteq NP \subseteq PSPACE$

X is PSPACE-complete $\wedge$ $X \in P$ $\Rightarrow$ $P = NP = PSPACE$

Randomized Algorithm: allow decisions based on coin flips

Algorithm A solves X with one-sided error if
$X(x) = YES \to A(x) = YES$ with prob $p > 0$
$X(x) = NO \to A(x) = NO$ with prob $= 1$
RP

two-sided error
$X(x) = YES \to A(x) = YES$ with prob $p > \frac{1}{2}$
$X(x) = NO \to A(x) = NO$ with prob $p > \frac{1}{2}$
BPP

Let $X \in BPP$ with alg A which gives correct answer with prob $\frac{3}{4}$

```
X(x)
  yes ← 0
  no ← 0
  for i=1 to n
      result ← A(x)
      if result = YES
          yes ← yes+1
      else
          no ← no+1
  if yes > no
      output YES
```

$BPP \stackrel{?}{=} P$ unknown

$BPP \subseteq NP$
$NP \subseteq BPP$ unknown

good enough for most

$n = 3 \qquad X(x) = YES$

$$P(\text{output}=YES) = \underbrace{P(2\text{ yes})}_{} + P(3\text{ yes})$$
$$= \left(\tfrac{3}{4}\right)^2 \cdot \left(\tfrac{1}{4}\right) \cdot 3 + \left(\tfrac{3}{4}\right)^3 = \frac{54}{64} > \frac{3}{4}$$

$n = 5$
$$P(YES) = \left(\tfrac{3}{4}\right)^3 \cdot \left(\tfrac{1}{4}\right)^2 \cdot 10 + \left(\tfrac{3}{4}\right)^4 \cdot \left(\tfrac{1}{4}\right) \cdot 5 + \left(\tfrac{3}{4}\right)^5$$
$$= \frac{918}{1024} > \frac{54}{64}$$

ZPP: decision problems with solutions with polynomial expected running time (but possibly unbounded worst case)

$$P \subseteq ZPP \subseteq NP$$

don't know if these are proper

(and if some NP-complete X satisfies $X \in ZPP$, then $ZPP = NP$)

$ZPP \subseteq NP$: Let $X \in ZPP$ [want $X \in NP$ – poly-time verification alg for X]

Then there is a expected poly-time algorithm A that solves X

Let p be polynomial s.t. the expected running time is $p(|x|)$

X-VERIFY(x, y) ← sequence of coin flips of length $p(|x|)$

simulate $A(x)$ using y as source of coin flips
return result

can't have all executions take > average time

$X(x) = YES \to$ some execution of $A(x)$ answers YES in $\leq p(|x|)$ time

$X(x) = \text{YES} \rightarrow$ some execution of $A(x)$ answers YES in $\leq p(|x|)$ time

$\rightarrow$ X-VERIFY(x, y) answers YES when $\underbrace{y \text{ is coin flips from}}_{\text{size } \leq p(|x|)}$ that execution

$X(x) = \text{NO} \rightarrow$ all executions of $A(x)$ answer NO

$\rightarrow$ X-VERIFY$(x,y) = \text{NO}$ for all y