NP search problem: (≈ NP-relation)
- goal: instance → solution (any)
- for each instance, set of (valid/feasible) solutions
- can recognize instances & their solutions in P

- every NP problem → NP search problem
  (for every choice of YES certificates → solutions)

**Counting version #A of NP search problem A**
- count number of solutions for given instance
- e.g. #SAT: find # satisfying assignments
  #Shakashaka: find # solutions to puzzle

#P = \{#A \mid \text{NP search problem } A \} \supseteq \text{count problems solved by polynomial-time nondeterministic counting algorithms} \supseteq \text{make guesses, at end says YES or NO (just like an NP algorithm)}

\[ \text{output } = \# \text{guess paths leading to YES} \]

#P-hard = as hard as all problems in #P via multicolor (Cook-style) reductions
⇒ #P unless P = NP
⇒ technically, FP = poly-time computable functions
Parsimonious reduction for NP search problems
- instance $x$ of $A$ $\rightarrow$ instance $x'$ of $B$
  - computable in polynomial time (like NP reduction)
  - $\#A$ solutions to $x$ = $\#B$ solutions to $y$
  $\Rightarrow$ decision problems (3 solution?) same answer
  $\Rightarrow$ NP reduction too

- $\#A$ is $\#P$-hard $\Rightarrow$ $\#B$ is $\#P$-hard
  ($& A$ is NP-hard $\Rightarrow$ $B$ is NP-hard)

C-monious reduction: uniform scaling
- $c \cdot \#A$ solutions to $x$ = $\#B$ solutions to $y$
  - preserves $\emptyset$ $\Rightarrow$ NP reduction too
  - $\#A$ is $\#P$-hard $\Rightarrow$ $\#B$ is $\#P$-hard

#P-complete SAT problems:
- $\#3SAT$
  - planar $\#3SAT$
    - planar monotone rectilinear $\#3SAT$
    - planar positive rectilinear $\#1$-in-$3SAT$
  - planar positive $\#2SAT-3$
  \{[Hunte II, Marathe, Radhakrishnan, Stearns - SICOMP 1998]\}
    \{as in L}\n
- Schaefer-style dichotomy:
  - $\#SAT \equiv FP$ $\Leftrightarrow$ system of linear equations (mod 2)
  - $\#SAT$ $\#P$-complete otherwise
  \{[Creignou & Hermann - I&c 2006]\}

see \{[Creignou, Khanna, Sudan - SIGACT 2001]\}
**Shakashaka**: parsimonious $\Rightarrow \#P$-hard  
[Demaine, Okamoto, Uehara, Uno - CCCG 2013]

**Hamiltonian cycles:**
- old proofs not parsimonious [Lichtenstein] [Plesnik]
- parsimonious reduction from 3SAT to planar max-degree-3 Hamiltonian cycle  
  [Sato - senior thesis 2002]
- nonplanar case solved earlier [Valiant 1974]

**Slitherlink**: parsimonious $\Rightarrow \#P$-hard  
- here can't use grid graphs  
  $\Rightarrow$ optional vertex gadgets  
  [Yato 2000]
Determinant of an \( \times \) \( n \) matrix \( A = (a_{ij}) \): 

\[
\text{permutation } \prod_{i=1}^{n} \frac{(-1)^{\text{sign}(\pi)}}{a_{i, \pi(i)}}
\]

Product of permutation matrix within \( A \)

Permanent: 

\[
\text{permutation } \prod_{i=1}^{n} a_{i, \pi(i)}
\]

\( \Rightarrow \) weighted directed \( n \)-node graph \( w(i,j) = a_{ij} \):

\[
\text{product of edge weights} \prod \text{cycle cover} \text{ of vertex-disjoint directed cycles hitting all vertices}
\]

- \#P-complete \cite{Valiant-TCS1979}
- c-monious reduction from \#3SAT
- weight-1 edges in variable & clause gadgets
- special weight matrix \( X \) in junctions
- \( \text{perm } X = 0 \Rightarrow \) not alone in nonzero cycle cover
  \( \Rightarrow \) entered & exited by bigger cycle
- \( \text{perm } (X - \text{row & col. } 1) = \text{perm } (X - \text{row & col. } 4) = 0 \)
  \( \Rightarrow \) can’t enter & leave immediately
  \( \Rightarrow \) enter at one end (1 or 4), leave at other
- \( \text{perm } (X - \text{rows & cols. } 1 & 4) = 0 \)
  \( \Rightarrow \) can’t leave interior 2x2 separate
  \( \Rightarrow \) must be visited between enter & exit
- \( \text{perm } (X - \text{row } 1 - \text{col. } 4) = \text{perm } (X - \text{row } 4 - \text{col. } 1) = 4 \)
  \( f \)actor for each traversal
- \( \Rightarrow \) acts as forced edge in var. & clause gadgets
- \( \text{perm } = 4^8 \cdot \# \text{clauses} \cdot \# \text{satisfying assignments} \)
Permanent $\mod r$ also \#P-hard: \cite{Valiant-TCS-1979}
- multical reduction from Permanent
- set $r = 2, 3, 5, 7, 11, \ldots$ until product $> M^n \cdot n!$
  - largest absolute entry in matrix $<$
  - $O(n \log M + n \log n)$ calls & max $r = O(\text{that ln that})$
- use Chinese Remainder Theorem \cite{Prime-\#-theorem}
  - encoded in unary

$0/1$-permanent $\mod r$:
- parsimonious reduction from permanent $\mod r$
  - all edge weights (effectively) nonnegative
- replace weight-$k$ edge ($k > 1$) with gadget with $k$ loops
  - unique solution if original edge unused
  - exactly $k$ solutions if original edge used (using exactly 1 loop)

$0/1$-permanent:
- one-call reduction from $0/1$-permanent $\mod r$
- call with same input
- return output $\mod r$

= \# cycle covers in given directed graph
= \# perfect matchings in given bipartite graph
  
  \((V_1 = \text{rows}, V_2 = \text{columns}, (i, j) \in E \iff a_{ij} = 1)\)

\(\overset{\nabla}{V_1} \overset{\nabla}{V_2}\)

(balanced: $|V_1| = |V_2|$)
Bipartite # maximal matchings: [Valiant - SICOMP 1979]
- one-call reduction from bipartite # perfect matchings
- replace each vertex with n copies (n=1!)
  & each edge with biclique K_{n,n}
⇒ old matching of size i
  → (n!)^i distinct matchings of size n_i
  (& preserves maximality)
- # maximal matchings in this graph
  = \sum_{i=0}^{n^2} (# orig. maximal matchings size i) \cdot (n!)^i
  ≤ (n^2)! e.g. K_{n/2,n/2}
⇒ can extract # perfect matchings (i = n/2)
incorrect (actually a reduction to another problem) fix:

Bipartite # matchings: [Valiant - SICOMP 1979]
- multicall reduction from bipartite # perfect matchings
- G → G_k: for each vertex: add k adjacent leaves
- M_r matchings of size n/2-r in G
  contained in M_r (k+1)^r matchings in G_k
⇒ # matchings in G_k = \sum_{r=0}^{n/2} M_r (k+1)^r
- evaluate this polynomial for k = 1, 2, ..., \sqrt{n/2} + 1
⇒ can extract coefficients M_0, M_1, ...
- M_0 = desired # perfect matchings in G

Bipartite # maximal matchings: [Vadhan - SICOMP 2001]
- one-call reduction from bipartite # matchings
- for each vertex: add 1 adjacent leaf
- matching → unique maximal matching
Positive \#2SAT

- \# vertex covers
- \# cliques in complement graph

- parsimonious reduction from bipartite \# matchings
- edge \rightarrow variable: true = not in the matching
- 2 incident edges e & f \rightarrow clause e \lor f
- satisfying assignment = matching

\# Minimal Vertex Covers

- \# maximal cliques in complement graph
- \# minimal truth settings for positive 2SAT
- parsimonious reduction from bipartite \# maximal matchings, as above
- minimal satisfying assignment = maximal matching
  \[ |E| - i \text{ true variables} \leq \text{ size } i \]

3-regular bipartite planar \# Vertex Cover

- planar positive 2SAT-3
  where each clause has 1 red & 1 blue variable
- \#P-complete [Xia, Zhang, Zhao - TCS 2007]

(2,3)-regular bipartite \# Perfect Matchings

- \#P-complete [Xia, Zhang, Zhao - TCS 2007]

(note: decision versions easy)
Another Solution Problem (ASP) [Ueda & Nagao - TR 1996]

- for NP search problem A:
  
  **ASP A:** given one solution, is there another?

- useful in puzzle design: want unique solution

- e.g. ASP k-coloring ∈ P (rotate colors)
  
  & ASP 3-regular Hamiltonian cycle ∈ P
  
  (always another solution)

**ASP reduction:** parsimonious reduction $A \rightarrow B$

& poly.-time bijection between solutions$_A(x)$ & solutions$_B(x')$

- includes every parsimonious reduction we've seen

$\Rightarrow$ ASP $A \rightarrow$ ASP $B$ via NP reduction

(can map given solution to $A \rightarrow$ sol. to $B$)

- ASP $B \in P \Rightarrow$ ASP $A \in P$

- ASP $A$ NP-hard $\Rightarrow$ ASP $B$ NP-hard

**ASP-hard =** ASP reducible from every NP search prob.

$\Rightarrow$ NP-hard & (c-)ASP problem is NP-hard

#solutions given, want another [Yato & Seta 2003]

**ASP-complete =** ASP-hard NP search problem

- includes planar 3SATs & Hamiltonicity today
  
  Shakashaka, Slitherlink

- not c-monious reductions e.g. 2SAT, matchings, permanent