6.046	Lecture 16	April 9, 2015

TODAY: NP-completeness
- NP-hardness & -completeness
- 3SAT
  - ↳ Super Mario Bros.
  - ↳ 3-Dimensional Matching
    - ↳ Subset Sum
      - ↳ Partition
        - ↳ Rectangle Packing
    - ↳ 4-Partition
      - ↳ Rectangle Packing
        - ↳ Jigsaw puzzles

weak (Subset Sum, Partition, Rectangle Packing)

strong (4-Partition, Rectangle Packing, Jigsaw puzzles)

Recall: (from 6.006)
- P = {problems solvable in polynomial time}
  - problems ↳ size n
  - polynomial ↳ $n^{O(1)}$
- NP = {decision problems solvable in polynomial nondeterministic time}
  - decision ↳ output is YES or NO
  - nondeterministic ↳ in O(1) time can "guess" among polynomial number of choices & if any guess leads to YES, then will make such a guess

"lucky"

- can assume all guessing is done first
$\Rightarrow$ equivalent to polynomial-time verifier of polynomial-size certificates for YES answers
- note asymmetry between YES & NO
- problem X is
  - NP-complete if $X \in NP$ & X is NP-hard
  - NP-hard if every problem $Y \in NP$ reduces to X
    - if $P \neq NP$ then $X \notin P$ ($NP \setminus P \to X$)

- reduction from problem A to problem B = polynomial-time algorithm converting A inputs into equivalent B inputs $A \to B$
  - same YES/NO answer
  - if $B \in P$ then $A \in P$ $\leftarrow A \to B \to$ solve
  - if $B \in NP$ then $A \in NP$
  - if A is NP-hard then B is NP-hard

How to prove X is NP-complete:

1. $X \in NP$ via nondeterministic algorithm or certificate + verifier
2. reduce from known NP-complete problem Y to X
   ($\Rightarrow$ any $Z \in NP \to Y \to X \Rightarrow X$ is NP-hard)
   - (A) poly-time conversion from Y inputs to X inputs
   - (B) if Y answer is YES then X answer is YES
   - (C) if X answer is YES then Y answer is YES

3SAT: given Boolean formula of the form:

$$(x_1 \underset{}{\overset{\text{OR}}{\vee}} x_3 \vee \overline{x_6}) \overset{\text{AND}}{\wedge} \underbrace{(\overline{x_2} \vee x_3 \vee \overline{x_7})}_{\text{clause}} \wedge \cdots$$

NOT

literals

i.e. formula = AND of clauses
clause = OR of 3 literals
literal $\in \{x_i, \overline{x_i}\}$ → occurrences of variable x_i

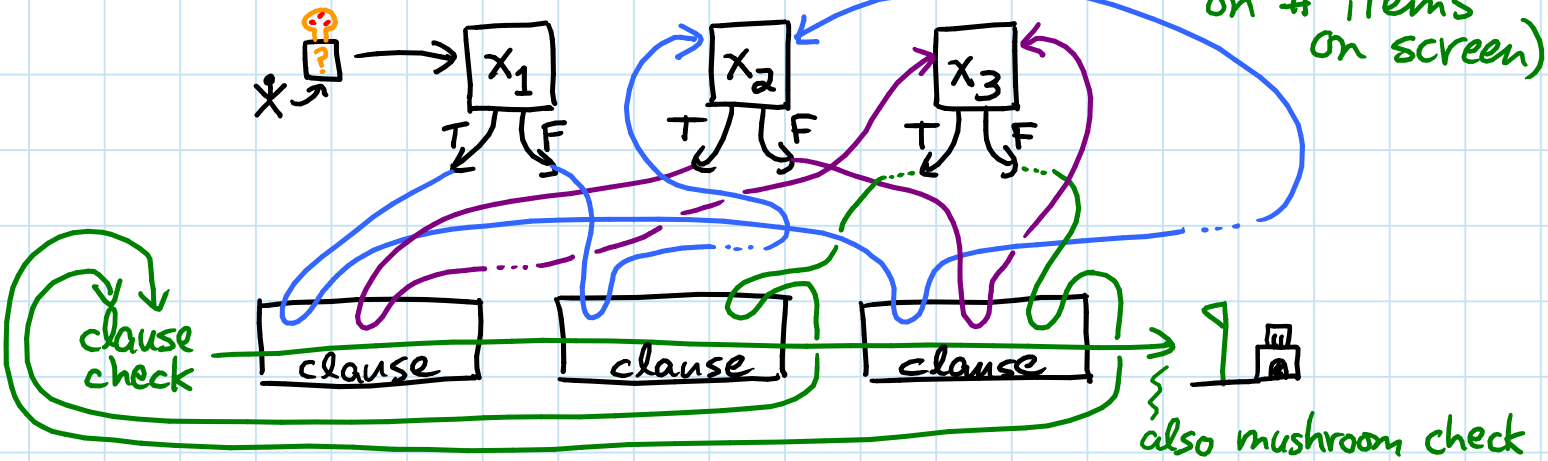
is there a variable → {T, F} assignment
such that formula = T (satisfying assignment)

- NP-complete [Cook 1971]
- $\in$ NP: guess x_1 is T or F
  guess x_2 is T or F
  ⋮
  } O(#variables) nondeterministic
  check formula — O(#clauses)
- NP-hard: intuition
  - convert algorithm into a circuit
  - convert circuit into a formula
  - convert formula into 3CNF (as above)

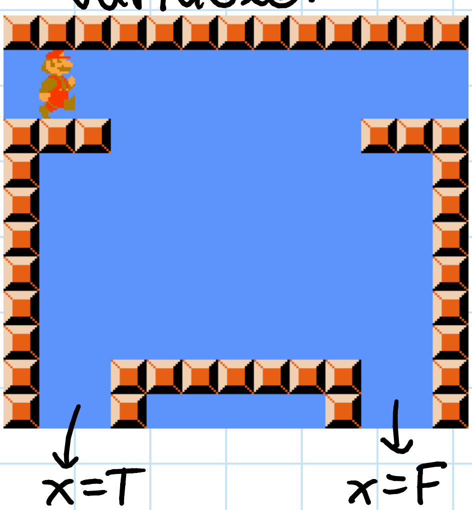
Super Mario Bros. is NP-hard

[Aloupis, Demaine, Guo, Viglietta 2014]

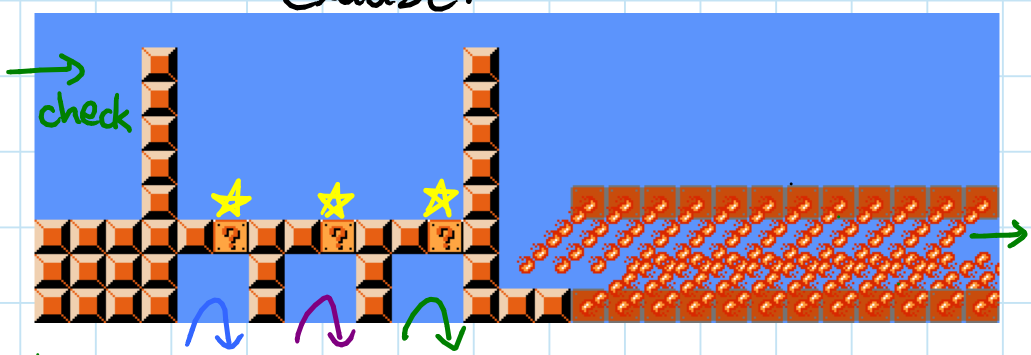
- generalized to arbitrary screen size ($n \times n$)
  (& remove limits on # items on screen)
- reduction from 3SAT:



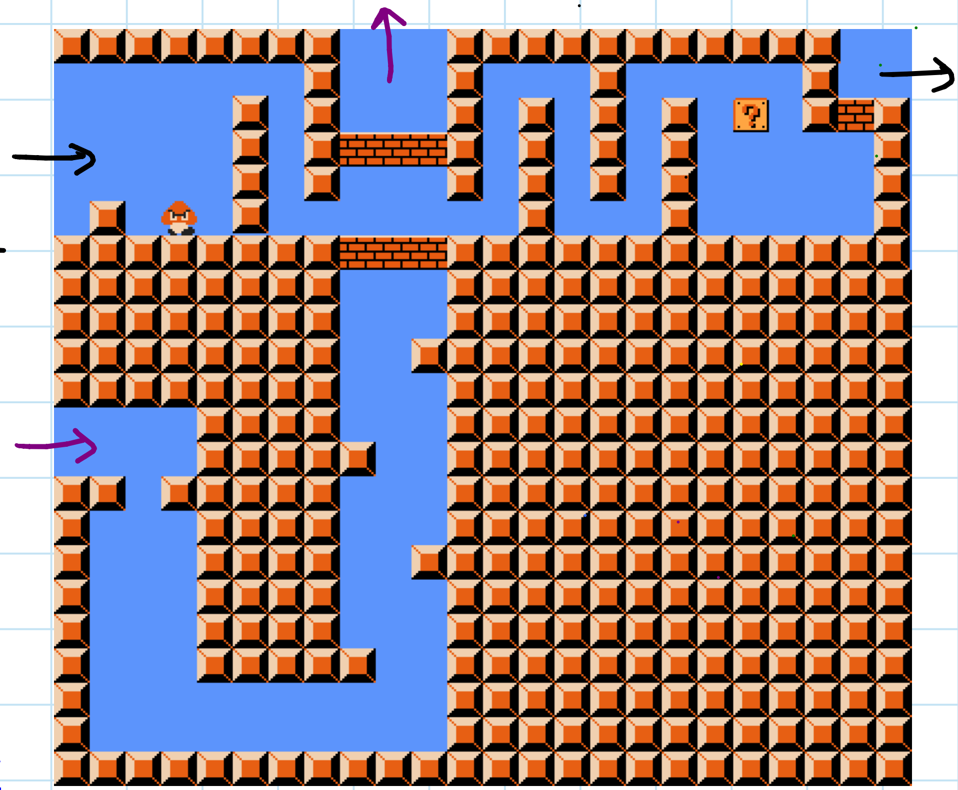
variable:



clause:



cross-over:



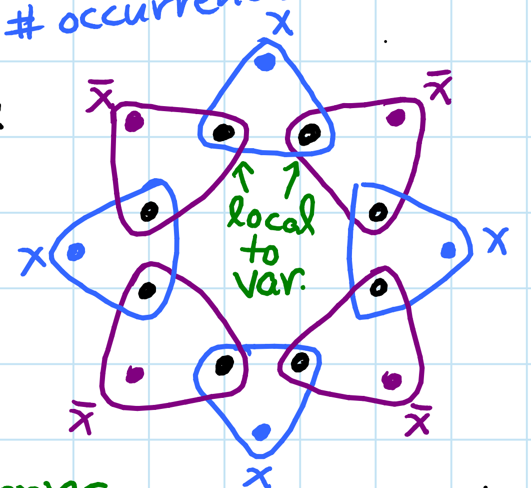
For many more cool examples, check out 6.890: "Fun with Hardness"

http://courses.csail.mit.edu/6.890/fall14/

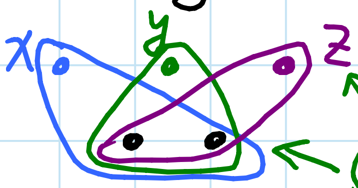
3-Dimensional Matching: (3DM)

given disjoint sets X, Y, Z each of n elements, & triples $T \subseteq X \times Y \times Z$, is there a subset $S \subseteq T$ such that each element $\in X \cup Y \cup Z$ is in exactly one $s \in S$?

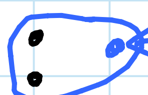
- $\in NP$: guess which triples $\in S$ — $O(T)$ nondet.
  check for exact coverage — $O(T)$
- NP-hard by reduction from 3SAT:
  [Garey & Johnson 1979 book]
  - variable $x \to 2n_x$ chain: (# occurrences of x or $\bar{x}$)
    - exactly 2 solutions
    - either x's or $\bar{x}$'s left



  - clause $x \vee y \vee z \to$



    - solvable if x or y or z's left
  - garbage collection:



    repeated $\underbrace{\sum_x n_x}_{\text{\# } x \text{ \& } \bar{x} \text{'s left by vars.}} - \underbrace{\text{\# clauses}}_{\text{\# covered by clauses}}$ times
    - can cover exactly all unused x_i's & $\bar{x}_i$'s
  - satisfying assignment $\to$ 3DM
    ($x = T \to$ leave x; $x = F \to$ leave $\bar{x}$; satisfy clauses; cover remaining with garbage collector)
  - 3DM $\to$ satisfying assignment
    (x left $\to x = T$; $\bar{x}$ left $\to x = F$; satisfy clauses)

Subset Sum: given n integers $A = \{a_1, a_2, \dots, a_n\}$
& a target sum t,
is there a subset $S \subseteq A$
such that $\sum S = \sum_{a \in S} a = t$?

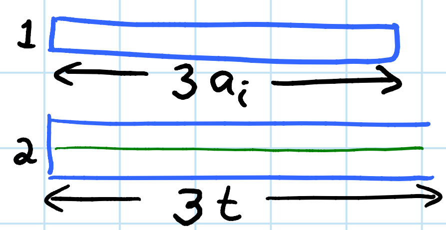
- $\in NP$: guess S
- pseudopolynomial algorithm via DP (like Knapsack)
  ↳ polynomial in n & sum of numbers (A)
- weakly NP-hard by reduction from 3DM
  ↳ hard when numbers exponential in n
  (but still only polynomial number of bits)
  - view numbers in base $b = 1 + \max_i n_{x_i}$
    #occurrences of x_i
    $\Rightarrow$ never overflow/carry
  - triple $(x_i, x_j, x_k) \to 000\underset{i}{1}00\underset{j}{1}0000\underset{k}{1}000$
    $= b^i + b^j + b^k$
  - $t = 11\cdots1 = \sum_i b^i$

Partition: given n positive integers $A = \{a_1, a_2, \ldots, a_n\}$, is there a subset $S \subseteq A$ with $\Sigma S = \Sigma (A \setminus S) = \frac{1}{2} \Sigma A$?

- special case of Subset Sum $(t = \frac{1}{2} \Sigma A)$
  $\Rightarrow \in NP$ & pseudopolynomial algorithm
- weakly NP-hard by reduction from Subset Sum
  - let $\sigma = \sum_{i=1}^{n} a_i$
  - add $a_{n+1} = \sigma + t$ & $a_{n+2} = 2\sigma - t$
    $\Rightarrow$ exactly one is $\in S$ (else 3σ vs. σ)
  - $\Rightarrow$ partition must add t to a_{n+2}
    & add $\sigma - t$ to a_n

Rectangle packing: given n integer rectangles $R_1, R_2, \ldots, R_n$ & target rectangle T of area $\sum_i \text{area}(R_i)$ can you pack R_i's into T without overlap?

- $\in NP$ because areas match
  $\Rightarrow$ can only rotate by int. $\times 90°$
  $\Rightarrow$ can guess rotation & integer translation
- weakly NP-hard by reduction from Partition:
  - $a_i \to 1 \times 3a_i$ rectangle R_i
  - $t \to 2 \times 3t$ rectangle T
  - $3 > 2 \Rightarrow$ can't rotate $90°$
  - $\Rightarrow$ packing must find partition



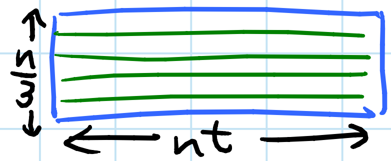
4-Partition: given n integers $A = \{a_1, a_2, \dots, a_n\}$, $\in (t/5, t/3)$
is there a partition into $n/4$ subsets of 4
each with the same sum $t = \Sigma A / (n/4)$?

also works with 3

- $\in NP$: guess $A \to$ subset mapping
- strongly NP-hard by reduction from 3DM [G&J]
  ↳ NP-hard even when number values polynomial in n
  - write numbers in base $r = 100 \cdot \Sigma(X \cup Y \cup Z)$
  - element $x_i \in X \to (10, i, 0, 0, 1) = 10r^4 + ir^3 + 1$
    & $(11, i, 0, 0, 1) \times (n_{x_i} - 1)$ copies
  - element $y_j \in Y \to (10, 0, j, 0, 2)$
    & $(11, 0, j, 0, 2) \times (n_{y_j} - 1)$ copies
  - element $z_k \in Z \to (10, 0, 0, k, 4)$
    & $(8, 0, 0, k, 4) \times (n_{z_k} - 1)$ copies
  - triple $(x_i, y_j, z_k) \to (10, -i, -j, -k, 8)$
    $= 10r^4 - ir^3 - jr^2 - kr^3 + 8$
  - target sum $t = (40, 0, 0, 0, 15) = 40r^4 + 15$
  - no carries (r large enough)
  - mod $r \Rightarrow$ use one x_i, one y_j, one z_k, one triple
  - $\lfloor \Sigma / r \rfloor$ mod $r \Rightarrow z_k$ & triple match
  - $\lfloor \Sigma / r^2 \rfloor$ mod $r \Rightarrow y_j$ & triple match
  - $\lfloor \Sigma / r^3 \rfloor$ mod $r \Rightarrow x_i$ & triple match
  - $\lfloor \Sigma / r^4 \rfloor$ mod $r \Rightarrow 4 \cdot 10$ → chosen triple $\in S$
    or $11 + 11 + 8 + 10$ → unused triple $\notin S$
  - primary (10) form of x_i (or y_j or z_k) must appear in exactly one chosen triple (and elements of triple must all match)

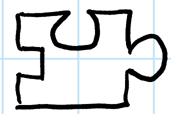
Rectangle packing:

- strongly NP-hard by reduction from 4-Partition
  - $a_i \to 1 \times n\,a_i$ rectangle R_i
  - $t \to \frac{n}{3} \times nt$ rectangle T

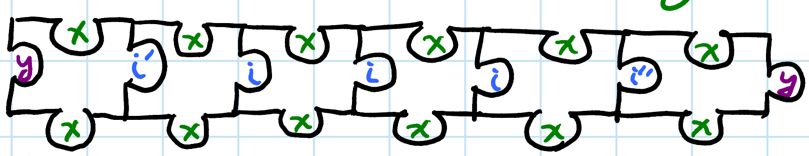


Jigsaw puzzles: [Demaine & Demaine 2007]

- model: square tiles (no pattern)
  each side tab, pocket, or boundary
  tabs & pockets must have matching shape
  target rectangular shape



- NP-hard by reduction from 4-Partition:
  (similar to reduction to Rectangle Packing)
  - $a_i \to$



  - $t \to$ frame: