Algorithmic Lower Bounds: Fun with Hardness Proofs

Super Mario Bros.

Crossover gadget for NP-hardness

Rush Hour

AND gadget for PSPACE-hardness

Minesweeper

OR gadget for NP-hardness

Hardness Made Easy*

Learn when to give up the search for efficient algorithms; see connections between computational problems; solve puzzles to prove theorems, solve open problems, and write papers.

Topics: NP, PSPACE, EXPTIME, EXPSPACE, 3SUM, approximation, fixed parameter, games & puzzles, key problems, gadgets, and proof styles.

6.892 taught by Professor Erik Demaine
11, AAGS, and Theoretical CS Concentration
Wednesday 7:00-9:30 pm in room 32-062
http://courses.csail.mit.edu/6.892/spring19/
sign up for our mailing list to join the class

Spring 2019

* Balance not guaranteed. Side effects such as open problems and a heightened sense of complexity may occur. Ask your advisor if 6.892 is right for you!
Complexity of Games & Puzzles

- **Simulation**: 0 players
- **Puzzle**: 1 player
- **Game**: 2 players (team, imperfect info)

Complexity Classes:
- **P**: Bounded
- **NP**: Bounded
- **PSPACE**: Bounded
- **EXPTIME**: Bounded
- **NEXPTIME**: Bounded
- **Undecidable**: Unbounded

Games: Rengo Kriegspiel? bridge?
Constraint Logic

[Hearn & Demaine 2009]

PSPACE  EXPTIME

Undecidable

PSPACE  NP  EXPTIME  NEXPTIME

P

0 players (simulation)  1 player (puzzle)  2 players (game)  team, imperfect info
Hamiltonian \((s, t)\)-Path

\[
\begin{array}{c}
\text{input}
\end{array}
\]

\[
\begin{array}{c}
\text{goal}
\end{array}
\]
100% Speedrun is NP-hard: Mario

The Lost Levels
Speedrun is NP-hard: Zelda
Speedrun is NP-hard: Metroidvania
Speedrun is NP-hard: RPG
Playing is NP-hard: Katamari
Edge-Matching Puzzles
[1890s—]
NP-hardness of $1 \times n$ Edge Matching

Bosboom, Demaine, Demaine, Hesterberg, Manurangsi, Yodpinyanee 2017

Reduction from Hamiltonian path with specified start $s$ & end vertex $t$
### NP-hardness of $1 \times n$ Edge Matching

<table>
<thead>
<tr>
<th>Start Vertex Tile</th>
<th>Vertex Tile</th>
<th>End Vertex Tile</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s = v_1$</td>
<td>$v_i$</td>
<td>$t = v_n$</td>
</tr>
</tbody>
</table>

**Tile Configurations**

- **Start Vertex Tile**
  - $A_{U_1}$
  - $O_{U_1}$
  - $v_1$

- **Vertex Tile**
  - $I_{U_i}$
  - $O_{U_i}$
  - $v_i$

- **End Vertex Tile**
  - $I_{U_n}$
  - $O_{U_n}$
  - $v_n$

**Edge Tile Configurations**

- **Edge Tile**
  - $O_{I_j}$
  - $X$
  - $(v_i, v_j)$

- **Path**
  - In path

- **Not in Path**
  - Not in path

---

**Remarks**

- The tiles represent different configurations in a graph matching problem.
- The symbols $A$, $U_i$, $O_i$, $U_n$, and $O_n$ represent various states or edges in the graph.
- The notation $s = v_1$ and $t = v_n$ indicates the start and end vertices, respectively.
NP-hardness

Bosboom, Demaine, Demaine, Hesterberg, Manurangsi, Yodpinyanee 2017