

# SAT Solver

Practice/Real-Life Applications of Computational Algorithms, Spring 2021

0710006 KE-YU LU

## Build

```
1 | make
2 | ./yasat [input_file_name]
```

## Environment

- Linux / macOS
- c++14

## Input Format

DIMACS

```
1 | c comment
2 | p cnf 3 4
3 | 1 2 3 0
4 | 1 -2 -3 0
5 | -1 2 -3 0
6 | -1 -2 3 0
```

## Output Foramt

Output the satisfiability and a solution we found!

```
1 | s SATISFIABLE
2 | v 1 2 3 0
```

```
1 | s UNSATISFIABLE
```

# Implementation

---

1. Store clauses in **Sparse Metrix** using STL vector
2. **Conflict-Driven Clause Learning (CDCL)**
3. **Boolean Constraint Propagation (BCP)**
  - **2-Literal Watching**
4. **Non-Chronological Backtracking**
  - **First Unique Implication Point (1UIP)**
5. Branching Heuristics
  - **Jeroslaw-Wang Score**
  - **Dynamic Heuristic**: Give higher priority to the new clauses.
  - Data Structure: STL set
6. **Luby Restart**