

NAME

nissy — a Rubik's cube solver and FMC assistant

SYNOPSIS

[-b]
nissy *command* [*options...*]

DESCRIPTION

nissy is a Rubik's Cube solver. Its optimal solver function uses Kociemba's one-step algorithm (huge optimal solver), and its performance is comparable to that of Kociemba's implementation in Cube Explorer when using a single CPU thread, and much faster when using multiple threads. **nissy** can also solve different sub-steps of the Thistlethwaite's algorithm and more.

When run without any argument an interactive shell is launched, otherwise the provided *command* is executed and **nissy** terminates. If the option **-b** is given, every argument after it is ignored and the shell is launched without any prompt or welcome message. This can be used to run **nissy** in batch mode, for example writing a list of commands in a *file* (one per line) and running **nissy -b < file**

The commands that can be run in the interactive shell are the same that can be run non-interactively and are provided below.

COMMANDS

The available *commands* are the following:

commands

List all available commands.

gen [-t N]

Generate all tables used by **nissy**. Run this to complete your installation. If *N* is specified, *N* CPU threads will be used (defaults to 1).

help [command]

Display help. If no *command* is given, a generic help message is printed, otherwise a specific help relative to *command* is returned.

print scramble

Display a text-only description of the cube obtained after applying *scramble*.

quit

Quit **nissy**.

solve step [options] scramble

Solve the given *step* on the given *scramble*. By default it finds only one (shortest) solution, without using **niss**, and it displays the number of moves at the end of the line. The options for the *solve* command are the following:

-a Print all solutions: some solutions are filtered out by default for some steps, for examples EOs that finish with F', with this options they are not.

-c Display only the number of solutions found, not the solutions themselves.

-m min

Only look for solution that are at least *min* moves long.

-M MAX

Only look for solution that are at most *MAX* moves long.

- n** Allow use of NISS.
- o** Only find solutions that require the minimum number of moves.
- p** Plain style: do not print the number of moves.
- s *n*** Try to find *n* solutions. By default and unless the **-M** or **-o** options are used, at most one solution is printed. If at least one of **-M** and **-o** is used, all the solutions found within the given bounds are printed. The option **-s o** overwrites these default behaviors and at most *n* solutions are printed, still satisfying the other constraints.
- t *n*** Use *n* CPU threads. By default nissy uses only 1 thread. Using more than one thread will improve performance, but the optimal number depends on your machine and operating system. Generally, using one less than the number of threads of your CPU works quite well.
- v** Verbose mode: print some information during the search and print each solution as it is found instead of only printing them all together at the end.

steps List all available *steps* for the *solve* command.

version

Display version information.

ENVIRONMENT

Data is stored in the folder pointed to by **\$NISSYDATA**. If that variable is unset the folder **\$XDG_DATA_HOME/nissy** or **\$HOME/.nissy** is used instead. If none of this environment variables is defined (e.g. in a non-UNIX system), the current folder is used.

EXAMPLES

The command:

```
nissy solve -v "R'U'FD2L2FR2U2R2BD2LB2D'B2L'R'BD2BU2LU2R'U'F"
```

Returns:

```
Found 0 solutions, searching depth 0...
Found 0 solutions, searching depth 1...
(some more lines)
Found 0 solutions, searching depth 16...
D2 F' U2 D2 F' L2 D R2 D F B2 R' L2 F' U' D
D2 F' U2 D2 F' L2 D R2 D F B2 R' L2 F' U' D (16)
```

The command:

```
nissy solve eofb -m 4 -M 5 -n -s 6 "R'U'FD2L2 FR2 U2R2BD2 L B2 D' B2
L' R'"
```

Returns:

```
U B U' B (4)
U2 D2 B (B) (4)
U2 F R2 F (4)
U2 B U2 B (4)
U2 B (U B) (4)
U' B U B (4)
```

AUTHORS

Sebastiano Tronto <sebastiano.tronto@gmail.com>

SOURCE CODE

Source code is available at <https://github.com/sebastianotronto/nissy>