hjweide / **a-star**

A very simple A* implementation in C++ callable from Python for pathfinding on a two-dimensional grid.

14 commits | **1** branch | **0** releases | **2** contributors | MIT

Branch: **master** | New pull request | Find file | Clone or download

hjweide Fix spacing. Latest commit `805df86` 21 days ago

mazes	First commit.	a year ago
solns	First commit.	a year ago
.gitignore	First commit.	a year ago
LICENSE	Initial commit	a year ago
Makefile	First commit.	a year ago
README.markdown	Optionally use 8-connectivity.	21 days ago
astar.cpp	Fix spacing.	21 days ago

examples.py	Optionally use 8-connectivity.	21 days ago
pyastar.py	Optionally use 8-connectivity.	21 days ago

README.markdown

A*

This is a very simple C++ implementation of the A* algorithm for pathfinding on a two-dimensional grid. The compiled `astar.so` file is callable from Python. See `pyastar.py` for the Python wrapper and `examples.py` for example usage. Uses 4-connectivity by default, set `allow_diagonal=True` for 8-connectivity.

Motivation

I recently needed an implementation of the A* algorithm in Python. Normally I would simply use networkx, but for graphs with millions of nodes the overhead incurred to construct the graph can be expensive. Considering that my use case was so simple, I decided to implement it myself.

Usage

1. Run `make` to build the shared object file `astar.so`.
2. Set the `MAZE_FPATH` and `OUTP_FPATH` as desired in `examples.py`.
3. Run `python examples.py`.

Example Results

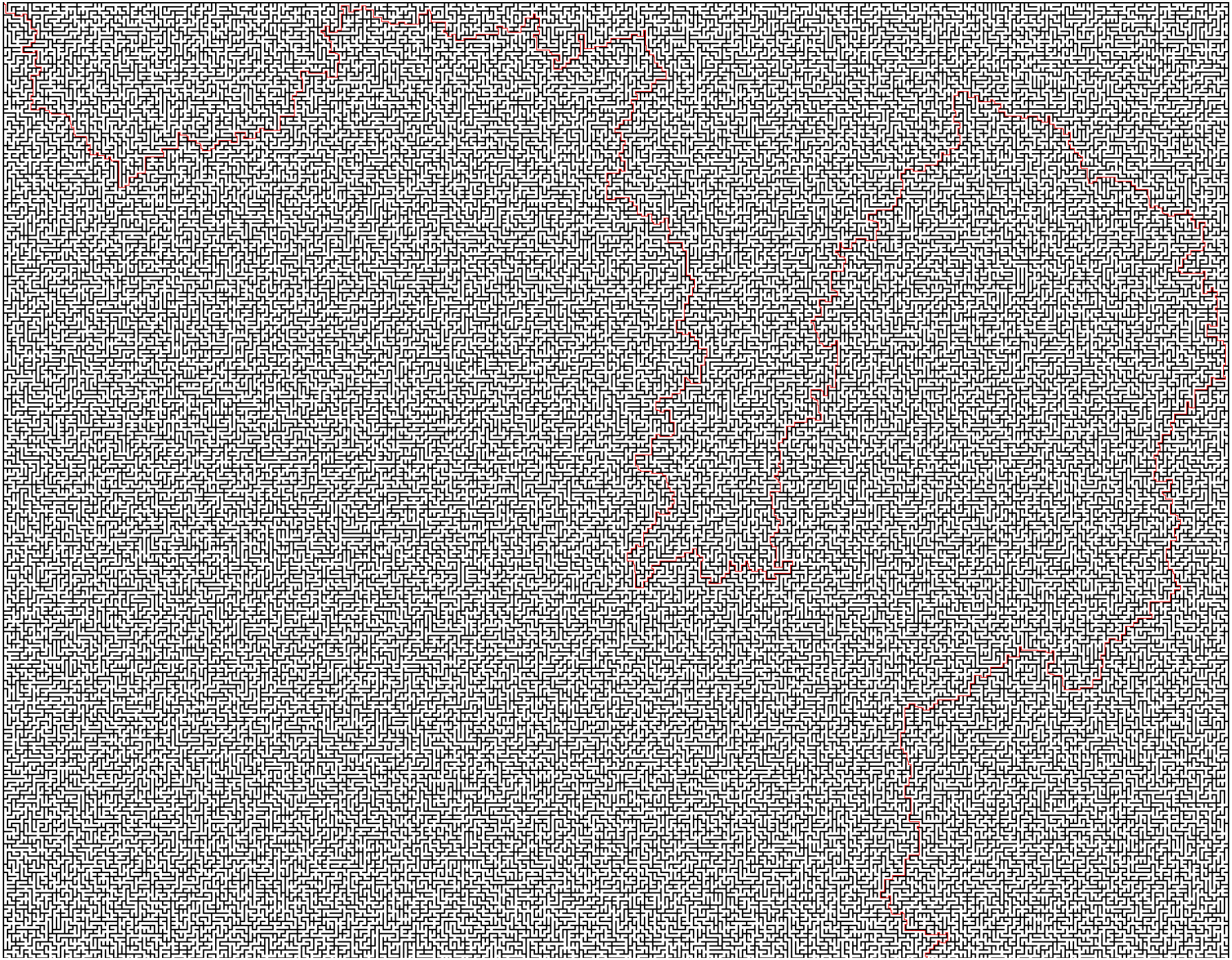
To test the implementation, I grabbed two nasty mazes from Wikipedia. They are included in the `mazes` directory, but are originally from here: Small and Large. I load the `.png` files as grayscale images, and set the white pixels to 1 (open space) and the black pixels to `INF` (walls).

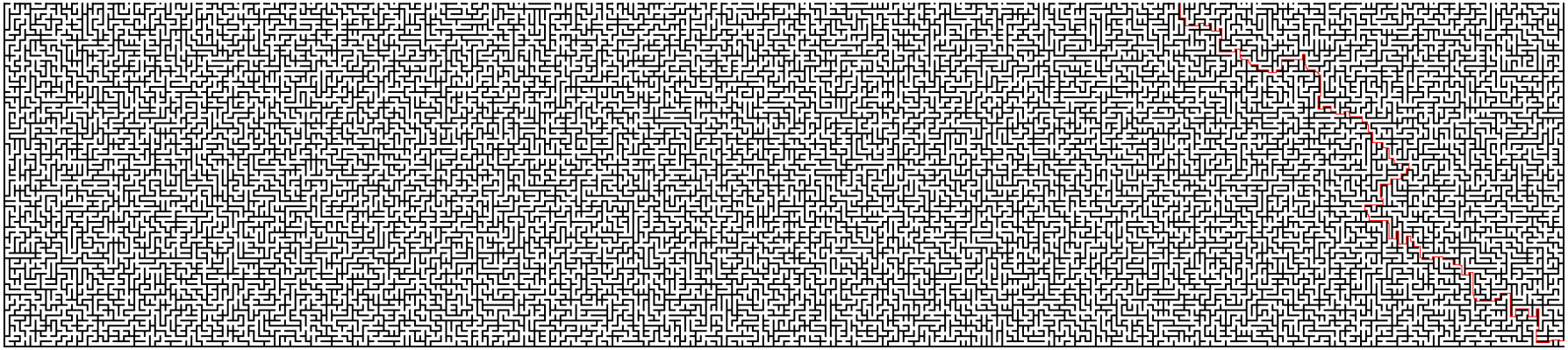
Run the code on the small maze:

```
time python examples.py
loaded maze of shape (1802, 1802)
found path of length 10031 in 0.258270s
plotting path to solns/maze_small_soln.png
done

real  0m2.319s
user  0m0.403s
sys 0m1.691s
```

The solution is visualized below:



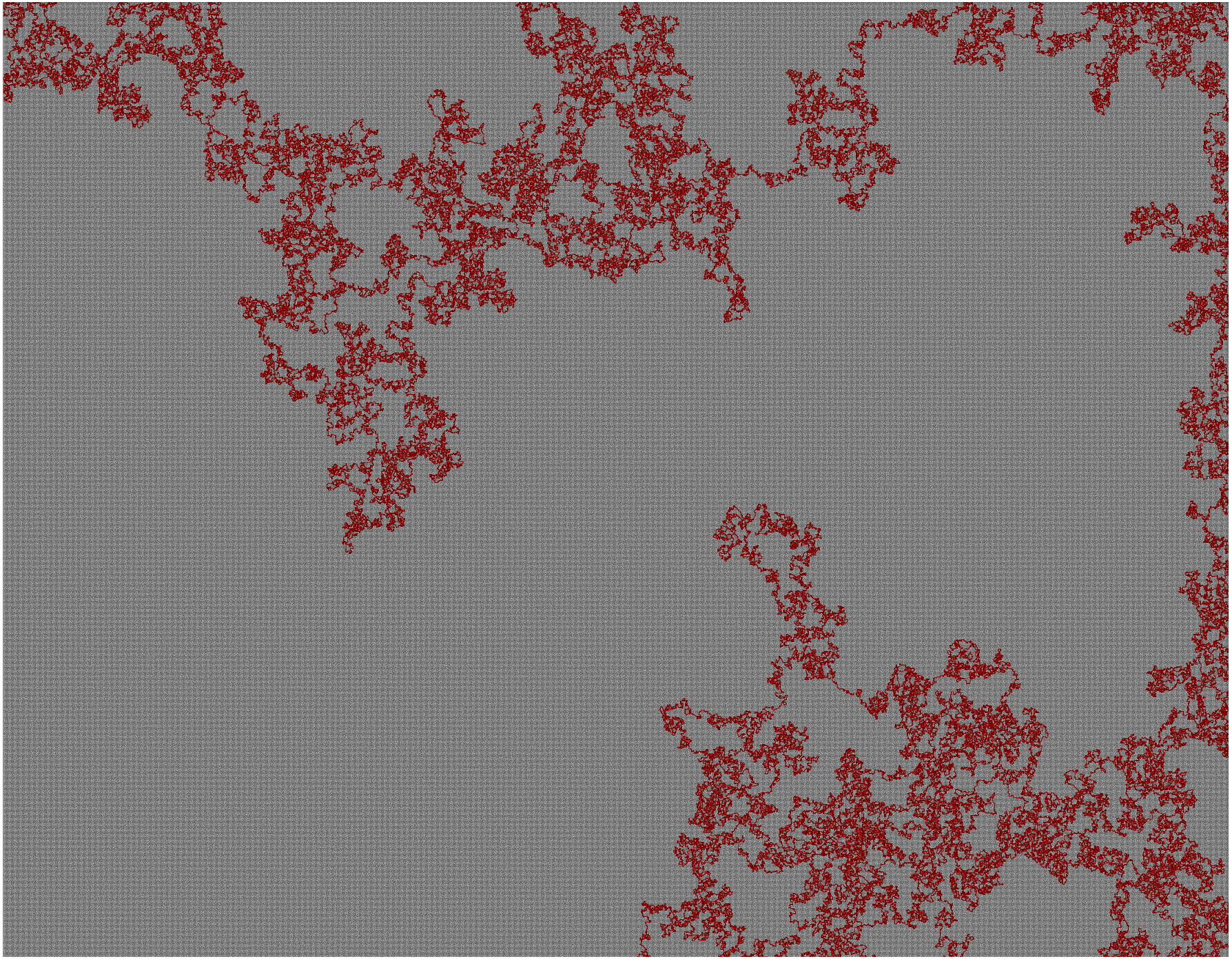


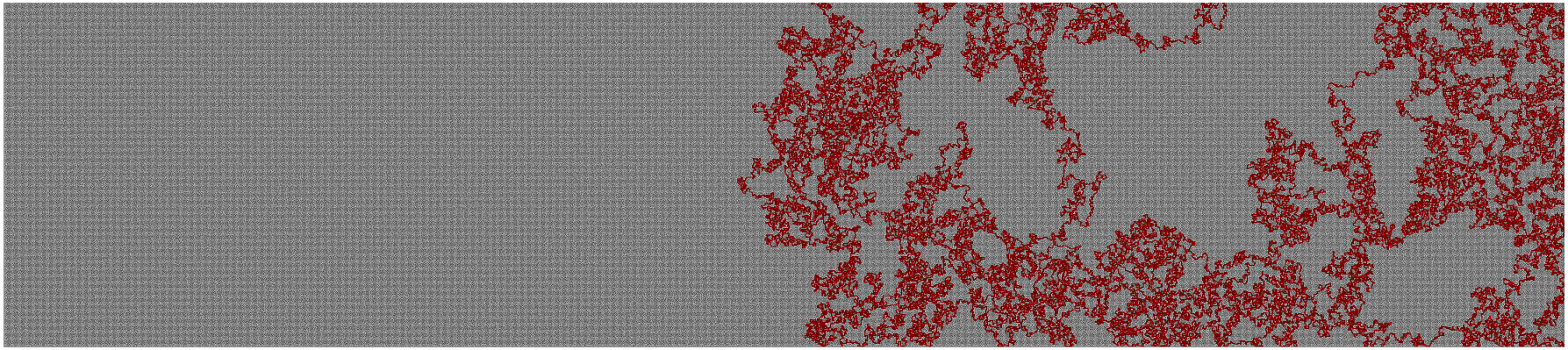
Run the code on the large maze:

```
loaded maze of shape (4002, 4002)
found path of length 783736 in 3.886067s
plotting path to solns/maze_large_soln.png
done

real  0m6.495s
user  0m4.007s
sys 0m2.273s
```

The solution is visualized below:





References

1. A* search algorithm on Wikipedia
2. Pathfinding with A* on Red Blob Games