UNIVERSITÀ
DEGLI STUDI
FIRENZE

Parallel Programming
Mid-term assignment

Lucia Giorgi

12/01/24

UNIVERSITÀ DEGLI STUDI FIRENZE

Index

UNIVERSITÀ DEGLI STUDI FIRENZE

Random Maze Solver

Find the exit from a maze using the random movement of a particle.

- Start a large number of particles, move them randomly bouncing on the walls
- Backtrack the first particle to get out of the maze to find the exit

UNIVERSITÀ DEGLI STUDI FIRENZE

Implementation

- Take an image of a maze from the Internet
- Extract information about the maze geometry and load it into an appropriate data structure
- Move the particles randomly from the starting point of the maze until it reaches the exit.

Cell

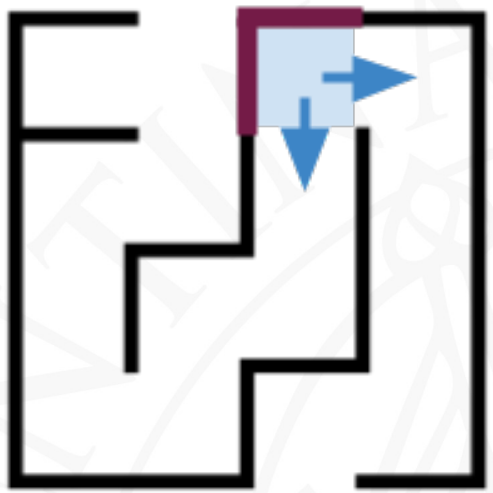


Figure: A cell with two possible directions and two walls

- (x, y) coordinates

UNIVERSITÀ DEGLI STUDI FIRENZE

Maze

- Move from a cell in a direction
- Start cell
- Load maze from image
- Save image with solution

Maze

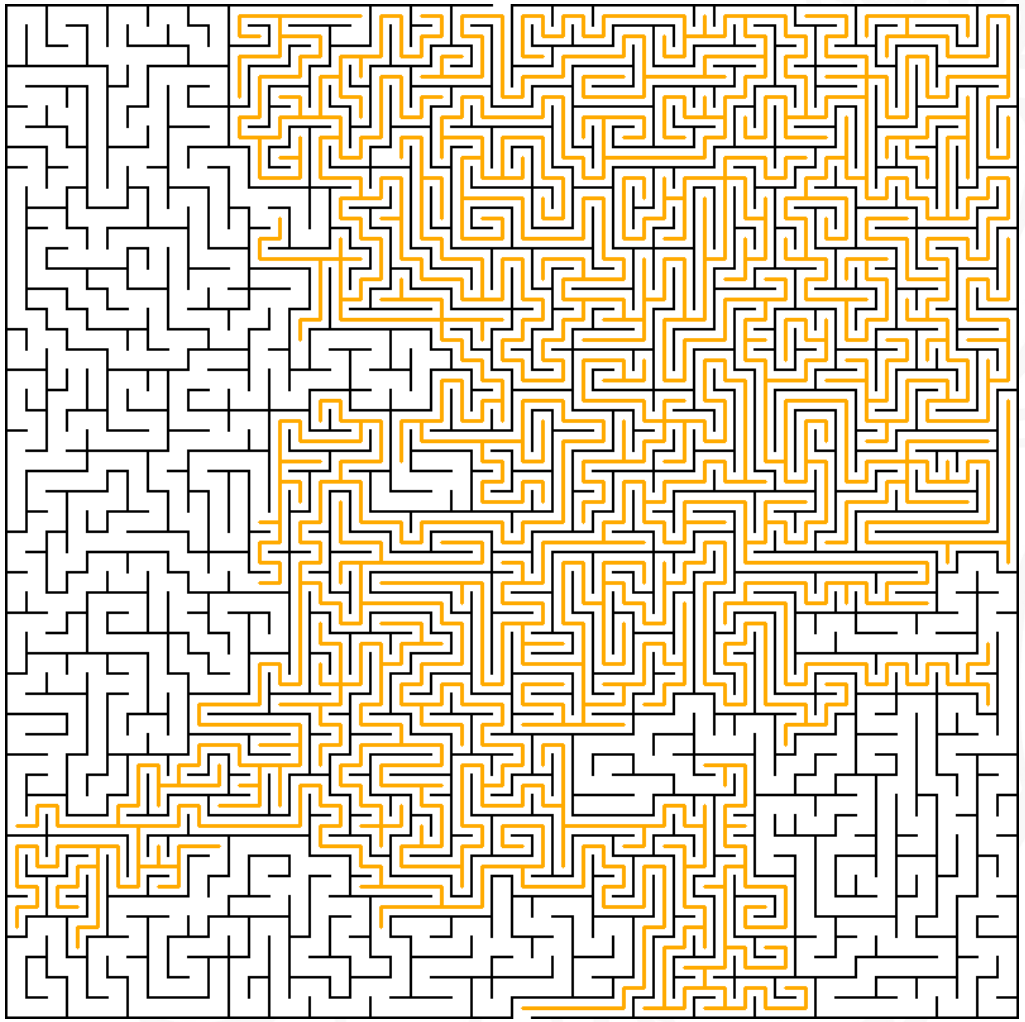


Figure: A 50 $\times$ 50 mazes with the solution found by the particle

Maze

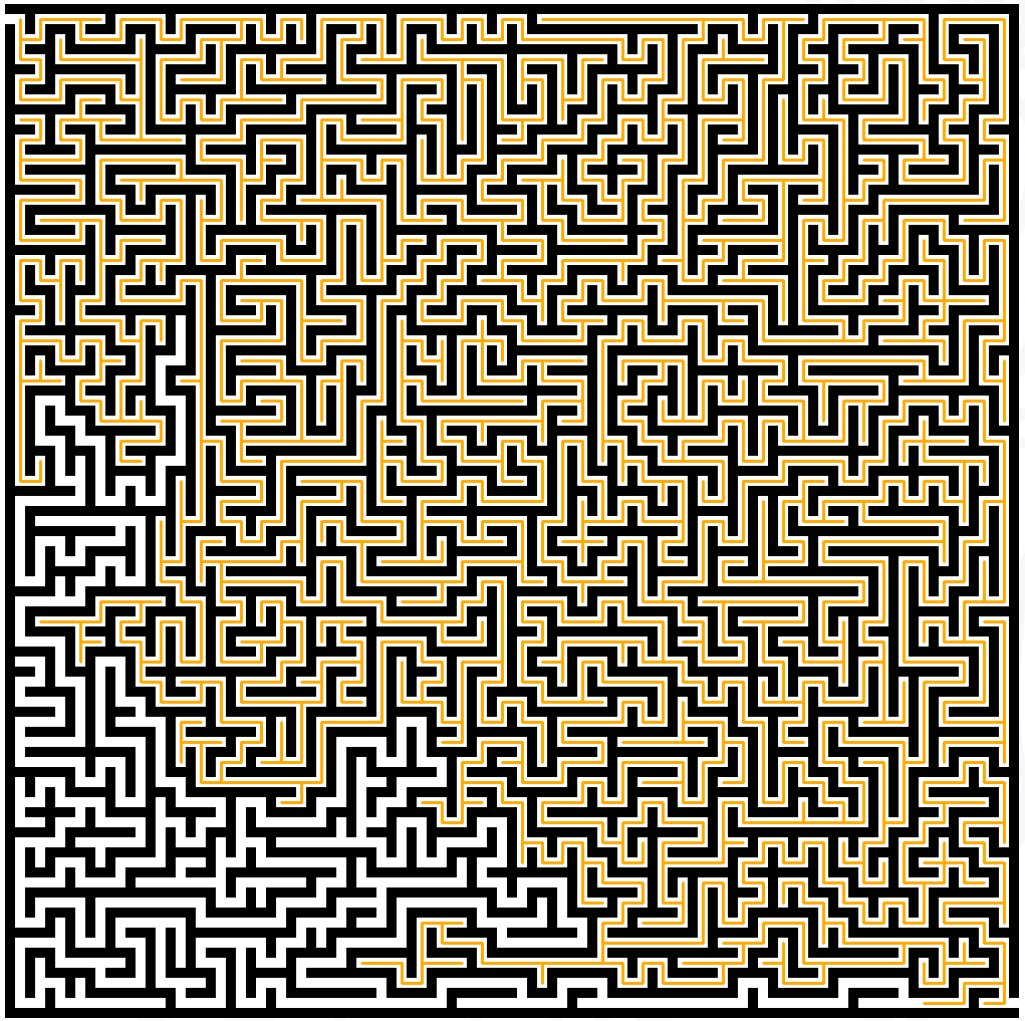


Figure: A 50 $\times$ 50 mazes with the solution found by the particle

Maze

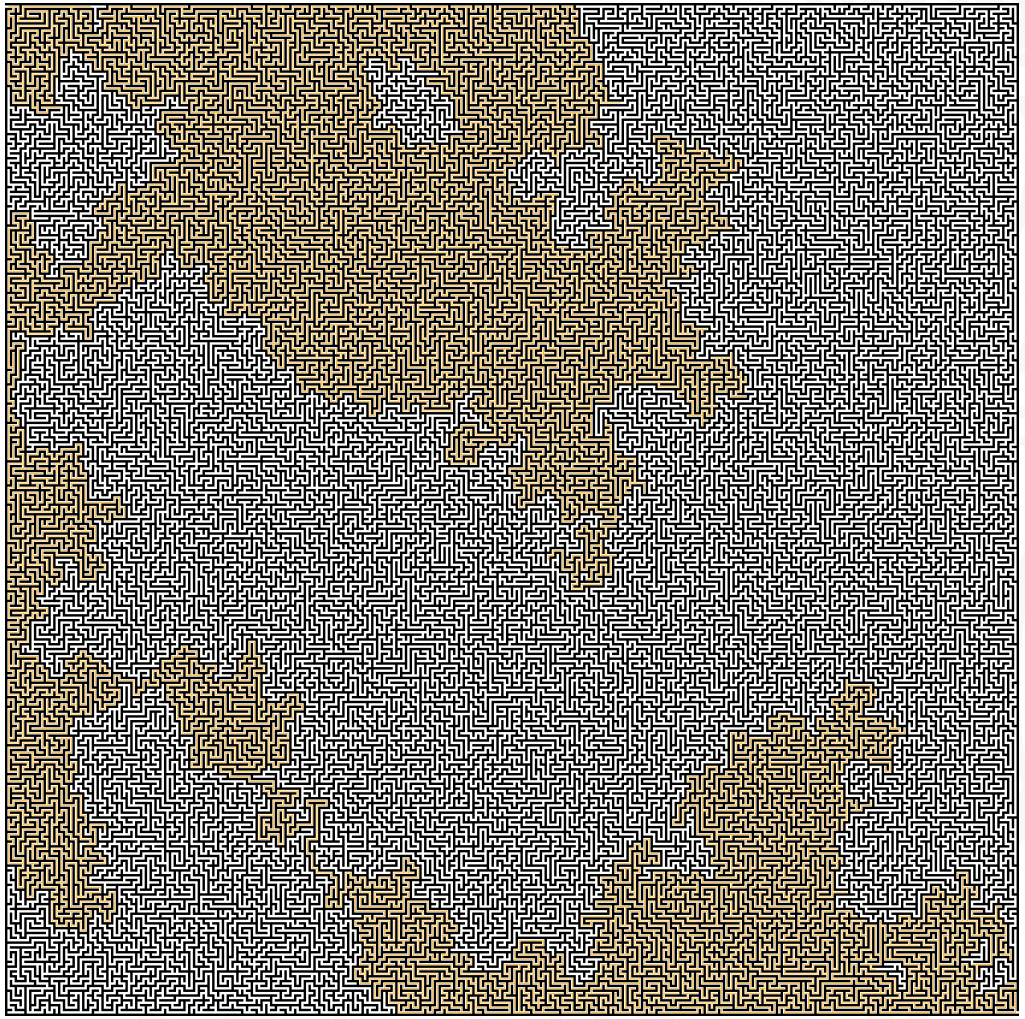


Figure: A 210 $\times$ 210 maze with the solution found by the particle

UNIVERSITÀ DEGLI STUDI FIRENZE

Main function

Four loops:

- Move particle to next cell as long as `solution_found==false`
- Loop over all particles
- Loop over all runs
- Loop over all images

UNIVERSITÀ DEGLI STUDI FIRENZE

Main function

Four loops:

- Move particle to next cell as long as `solution_found==false`
- Loop over all particles
- Loop over all runs
- Loop over all images

Command line arguments: particles number, thread numbers, runs, save solution

UNIVERSITÀ DEGLI STUDI FIRENZE

Parallelization

OpenMP

- `#pragma omp parallel for` on the particles loop
- `solution_found` is `shared`

Parallelization

```
while (!solution_found) {
// steps for moving the particle
// ...
     catch (OutOfMazeException &e) {
        solution_found = true;
        endTime = std::chrono::high_resolution_clock::now();
        bool is_first;
        #ifdef _OPENMP
        omp_set_lock(&solution_found_write);
        #endif
        if (!solution_found_locked) {
            solution_found_locked = true;
            is_first = true;
        }
        #ifdef _OPENMP
        omp_unset_lock(&solution_found_write);
        #endif
        if (is_first) {
            // log the execution time
            // ...
        }
    }
}
```

UNIVERSITÀ DEGLI STUDI FIRENZE

Experiments

- Sequential version
- Parallel version
  - 2, 4, 6, ... 30 threads
  - Particles number $\geq$ threads number

UNIVERSITÀ DEGLI STUDI FIRENZE

Experiments

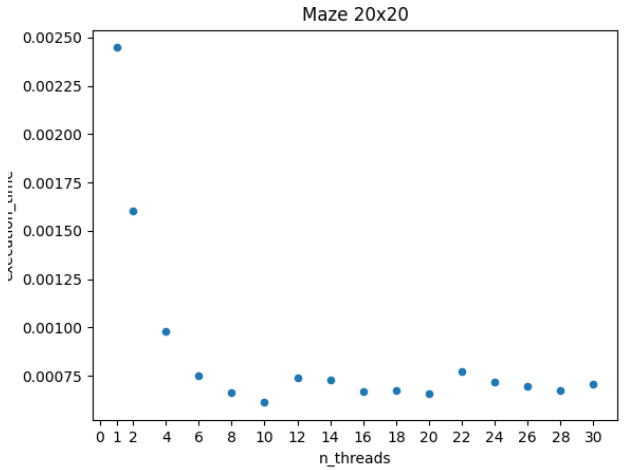
- Sequential version
- Parallel version
  - 2, 4, 6, ... 30 threads
  - Particles number $\geq$ threads number

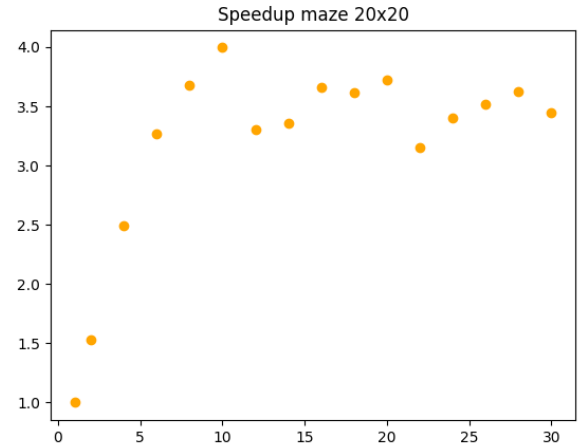
100 runs

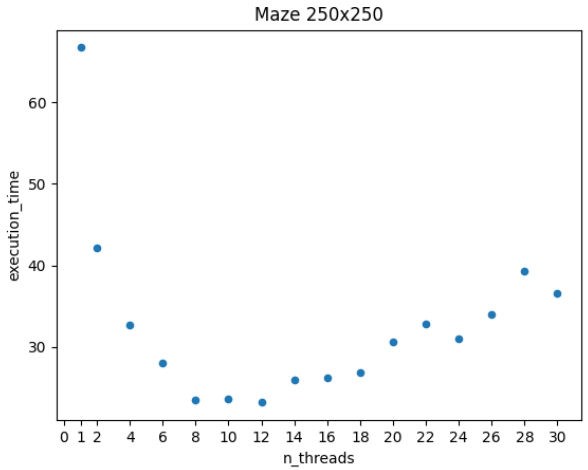
No solution

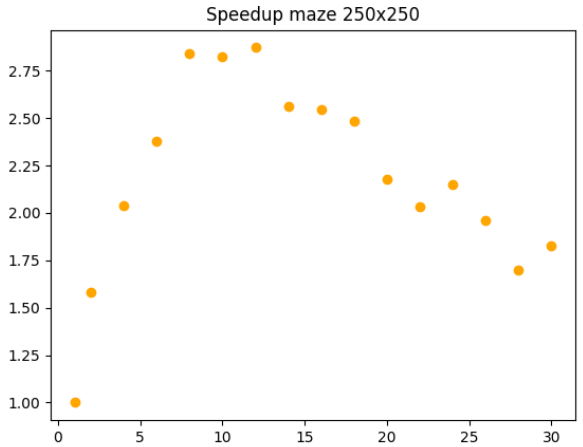
UNIVERSITÀ DEGLI STUDI FIRENZE

Results



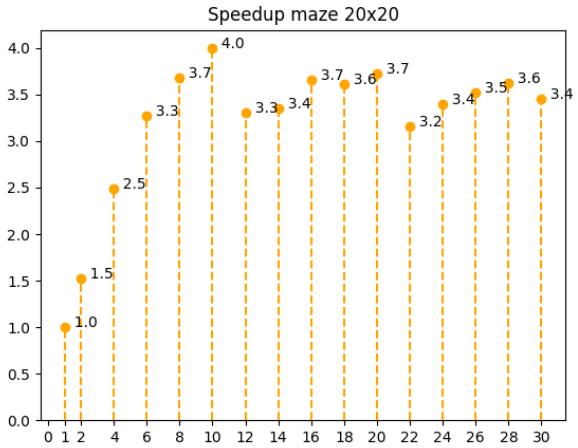


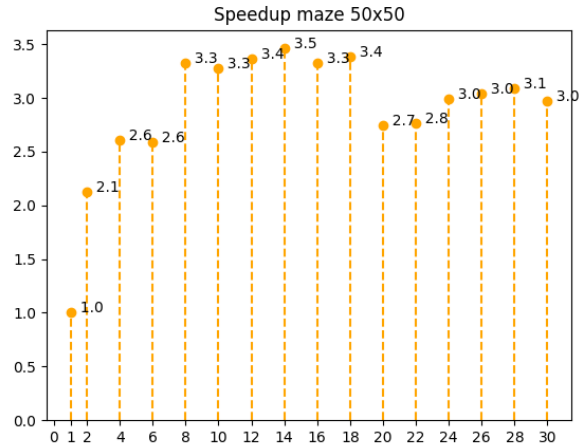


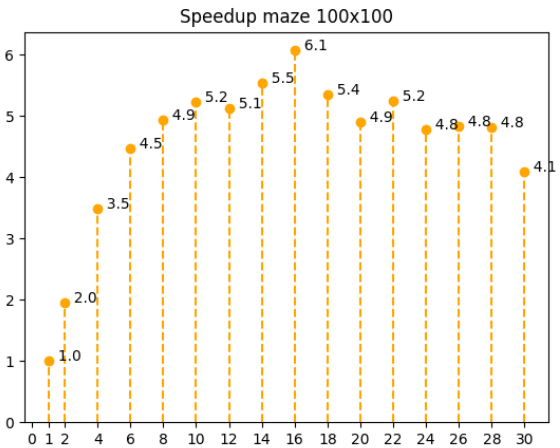


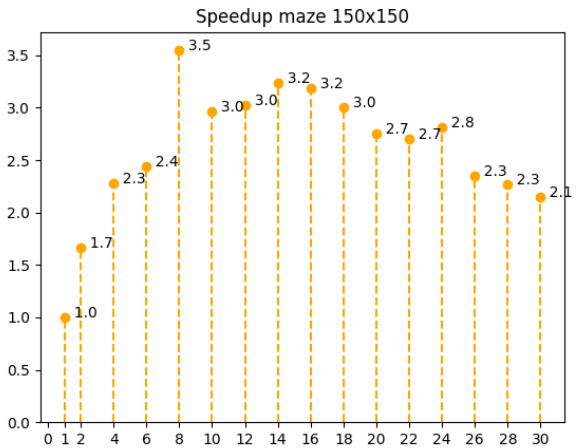
UNIVERSITÀ DEGLI STUDI FIRENZE

Results



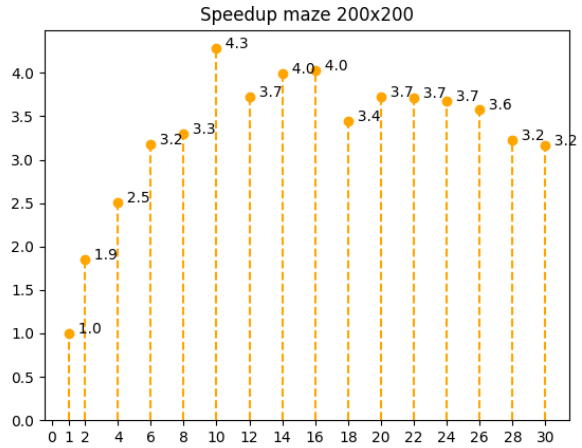


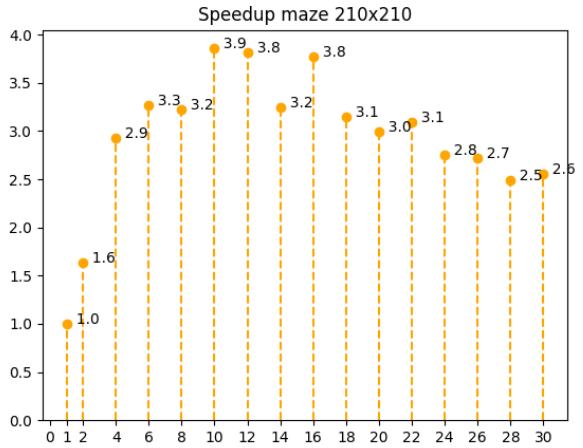


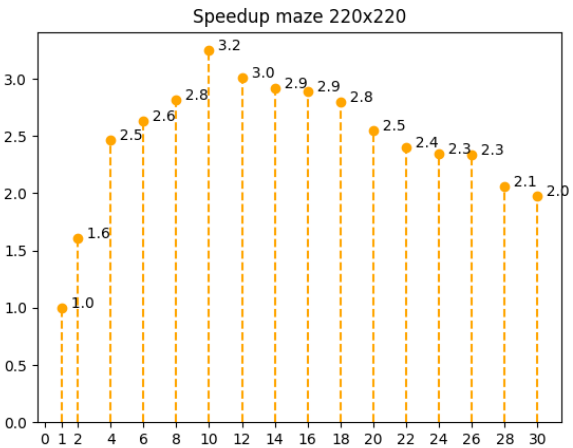


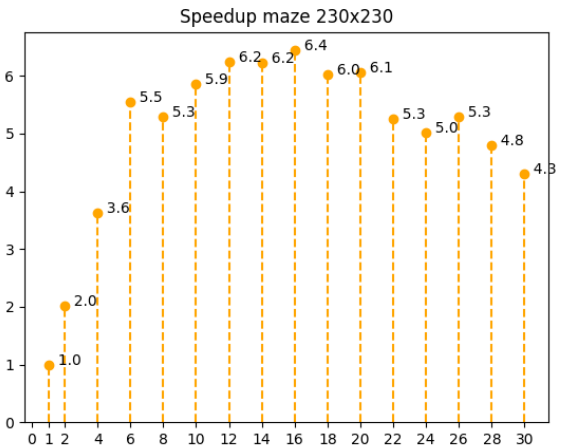
UNIVERSITÀ DEGLI STUDI FIRENZE

Results









UNIVERSITÀ DEGLI STUDI FIRENZE

Results