COMPARISON OF VARIOUS PATHFINDING ALGORITHMS

by DatPATH

Damla Büşra Çelik 150140102

Yunus Güngör 150150701

Korel Chairoula 150140910

What is our goal?

Why?

PROJECT TASKS

In this project, we tested 5 pathfinder algorithms with 3 different kind of mazes

- Depth First Search (DFS) Algorithm
- Breadth First Search (BFS) Algorithm
- A* Algorithm
- Dijkstra's Algorithm
- Random Algorithm

- Crooked Maze
- Flat Maze
- Crooked Combined with Flat Maze

Testing Environment

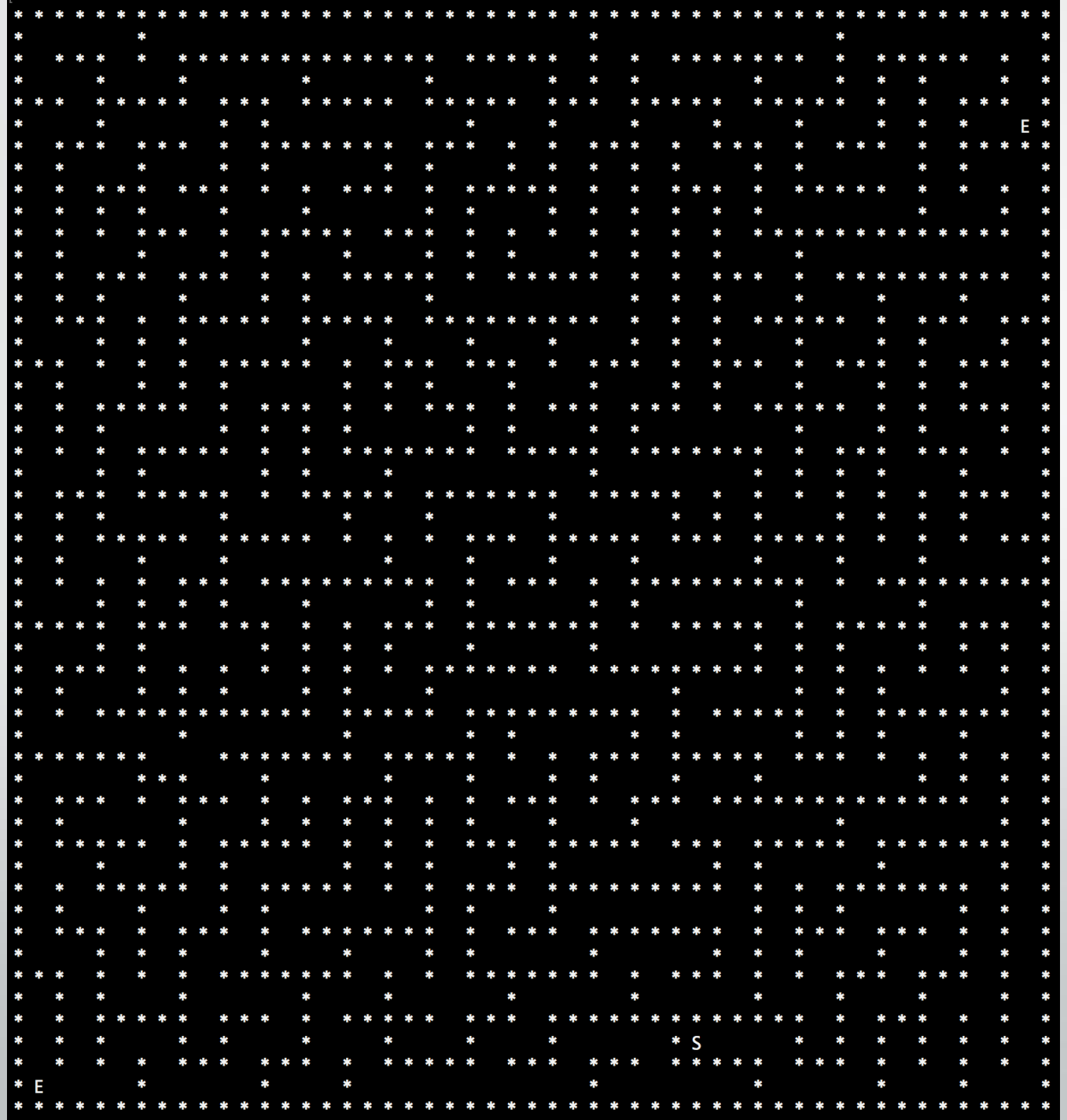
Computer / Model	CPU info	Max. Battery Cap.	Cur. Battery Cap.	Cur. Battery Health	*Battery Consumption in ≈1 min.	**Battery Cycle
Apple MacBook Pro (Mid 2015) (PC1)	2,8 GHz Intel Core i7 4980HQ L1: 32KB/32KB L2: 256KB L3: 6MB	8,755 mAh	8,095 mAh	≈92%	Avg. ≈14,2 mAh Avg. ≈0,159 mWh	228
Asus n n550jk cn167d (PC2)	2,4 GHz Intel Core I7 4700HQ L1: 256KB L2:1024KB L3:6MB	4000 mAh	2841.9594 mAh	≈71%	Avg. ≈32 mAh Avg. ≈0,473 mWh	No data

**** Battery Consumption in ≈1 minute:*** *The battery consumption with the default programs running in the OS (Operation System). In addition to that, mouse was plugged in, wi-fi was on and the screen brightness was full in order to get a standard consumption average. It was calculated right after a reboot several times to get the average.*

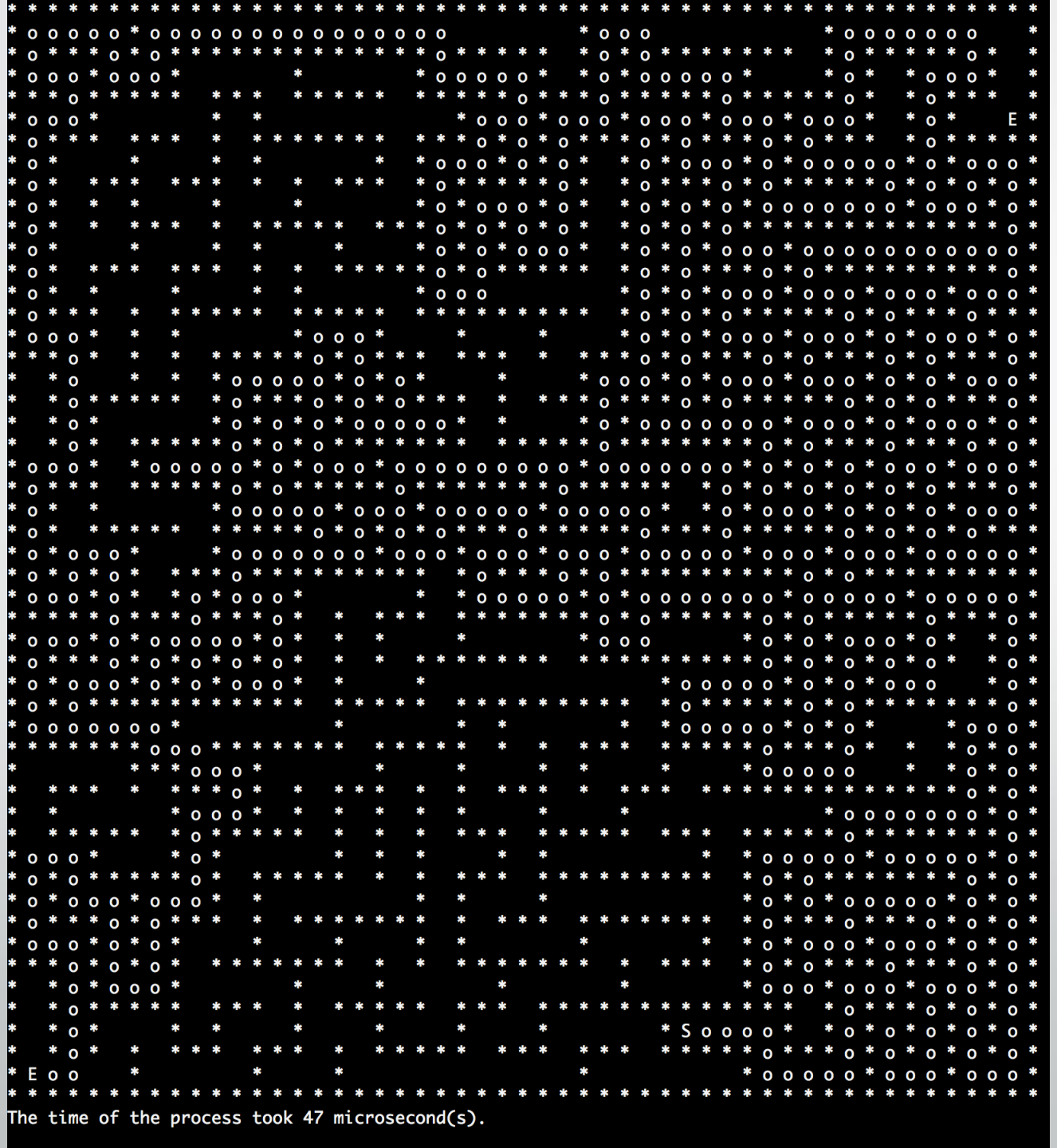
***** Battery Cycle:*** *The charge / discharge cycle in which shows the battery life.*

Implementation

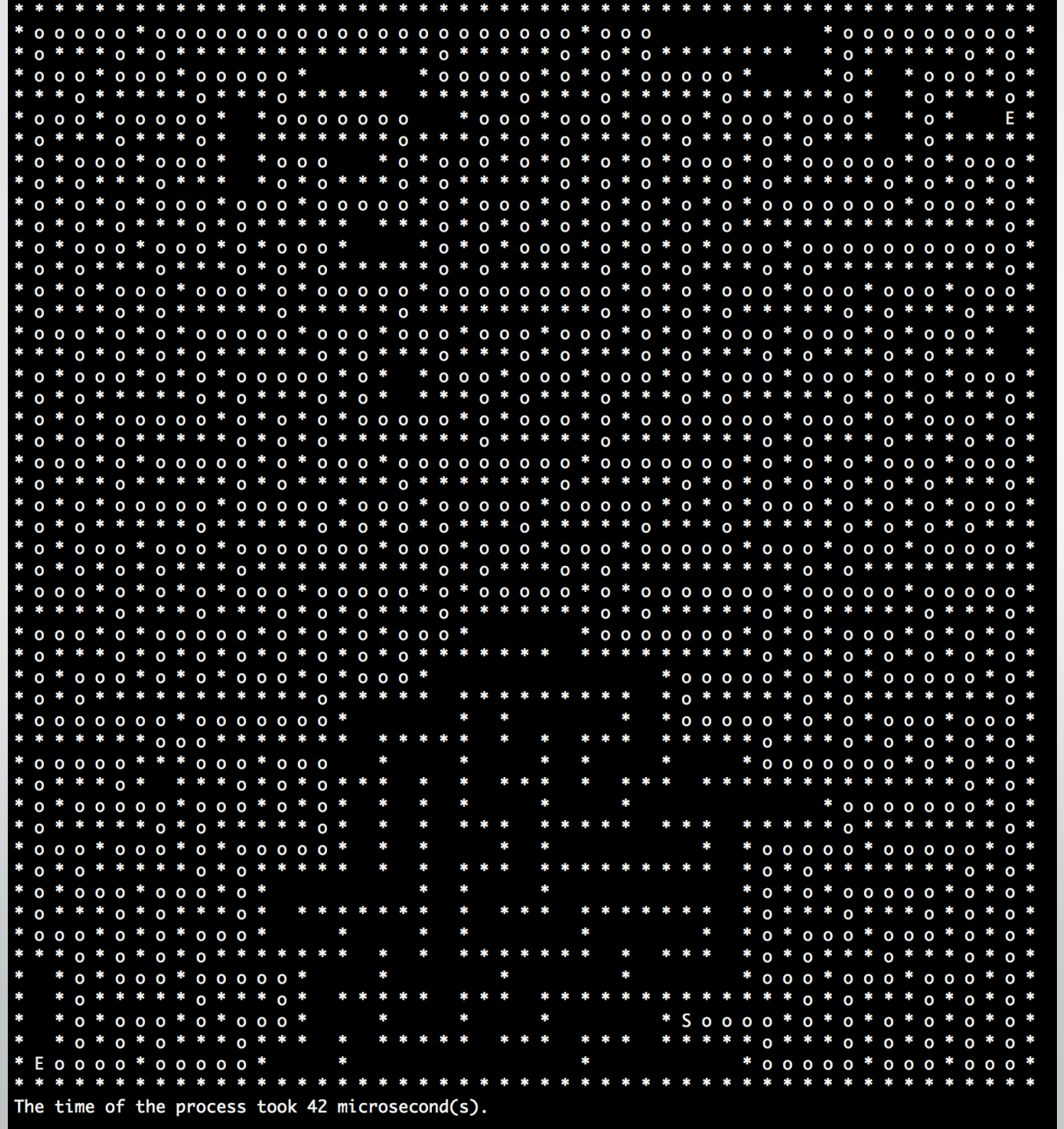
The Maze



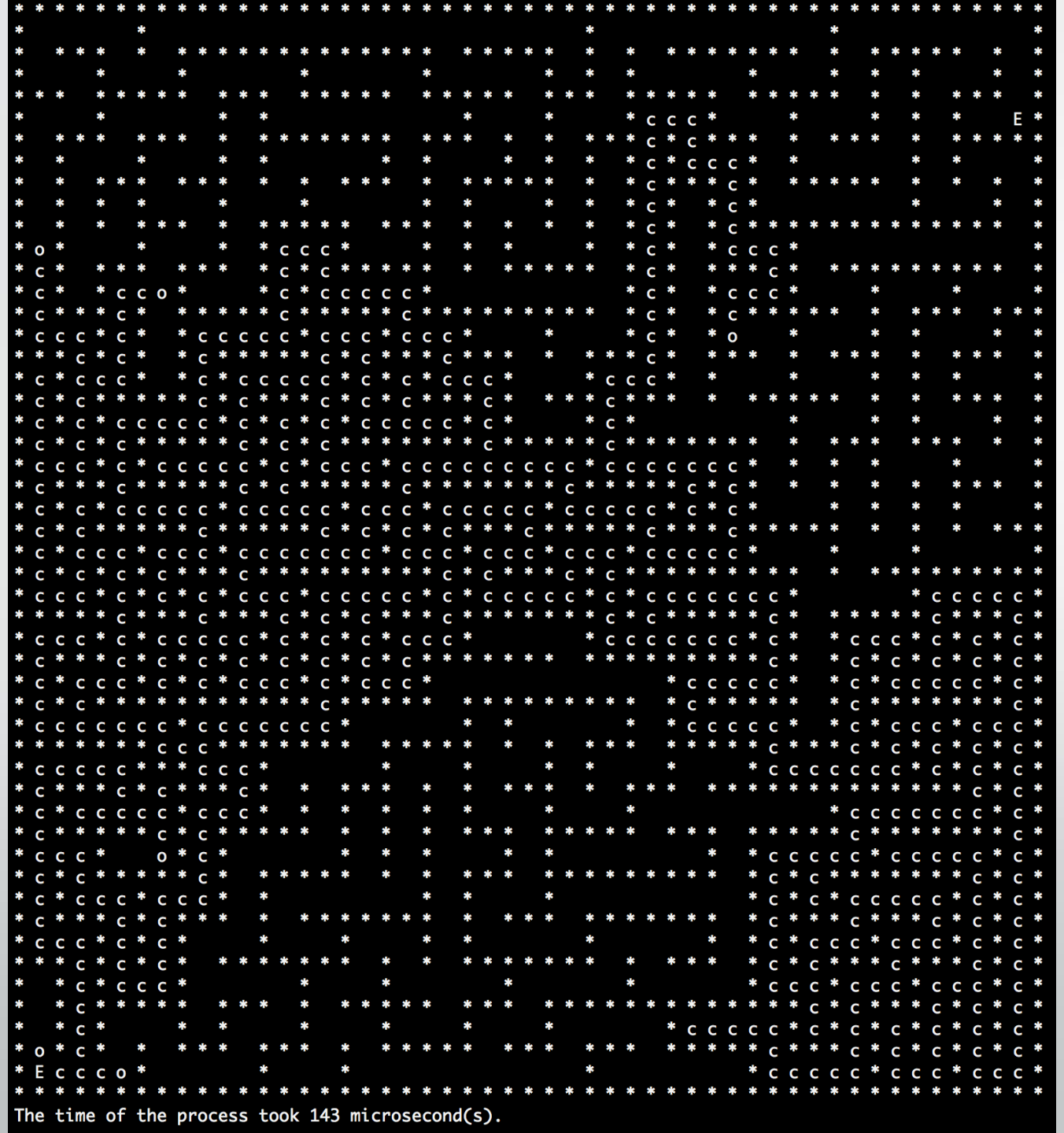
DFS



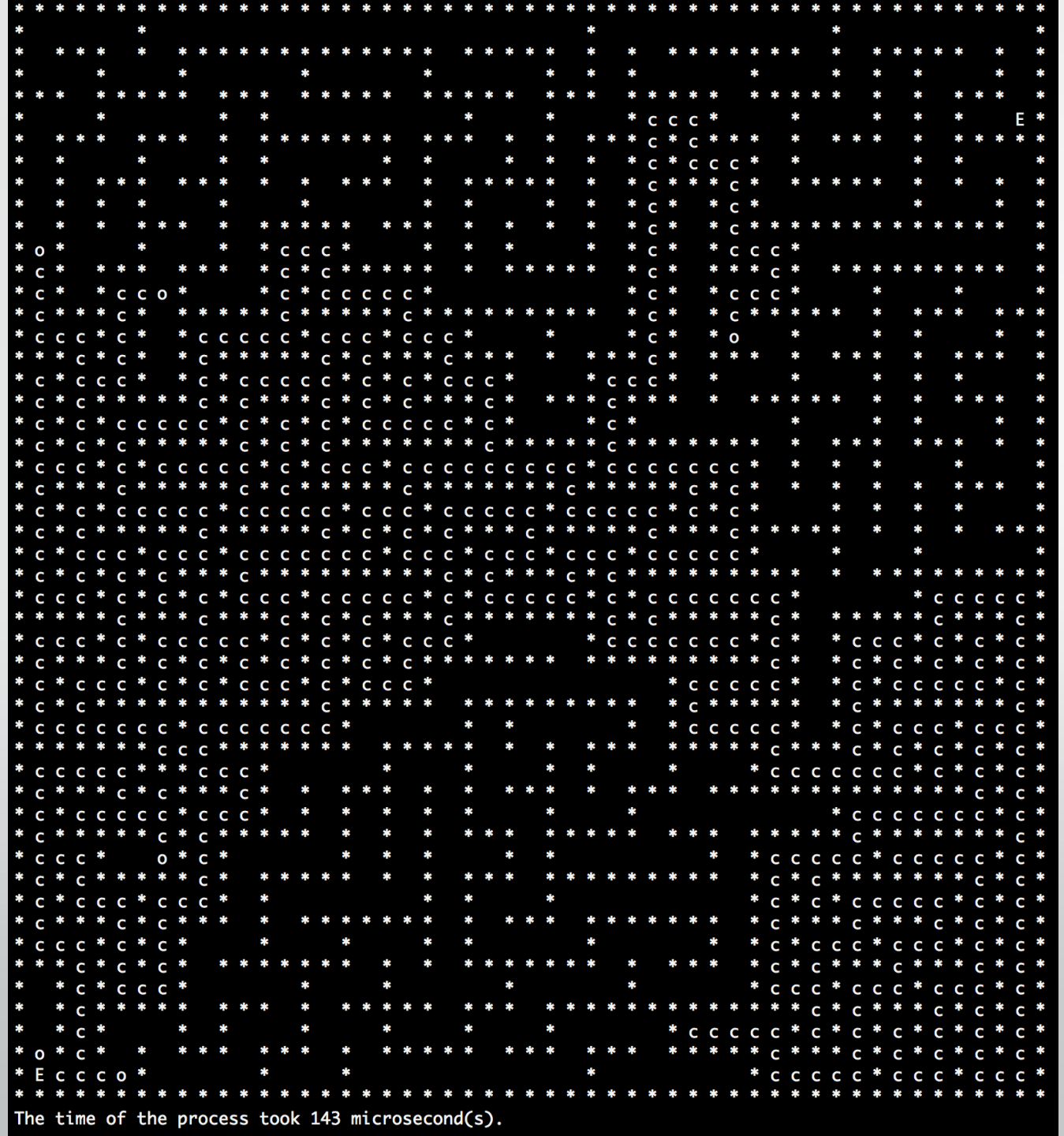
BFS



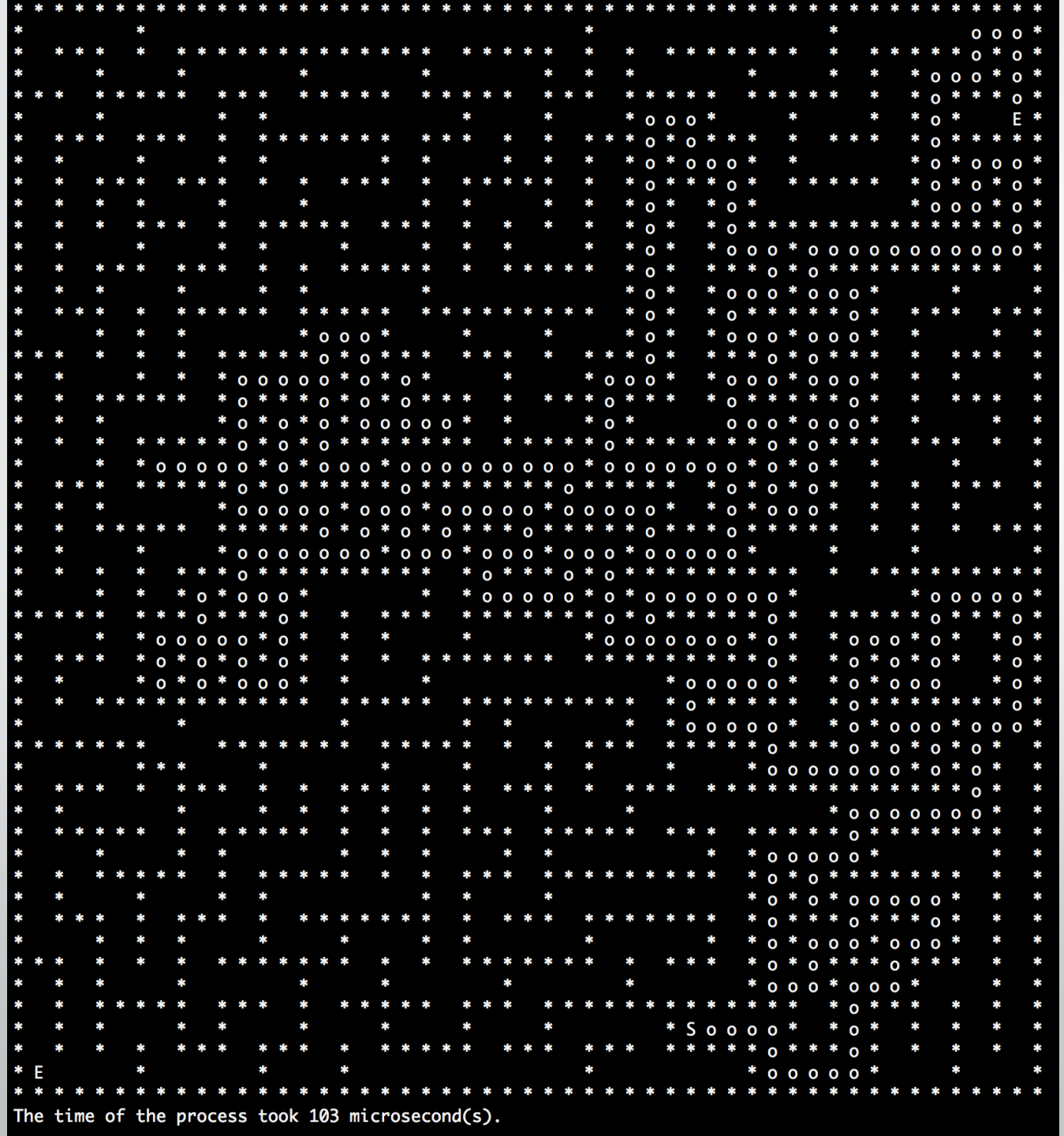
Dijkstra



A*



Random



Run Time Graphics

- Graphics' data taken from experiments done on PC1 and PC2. Experiment coded with C++. Those computers' specifications given in the previous table.

Figure 1-Total Run Time on PC1

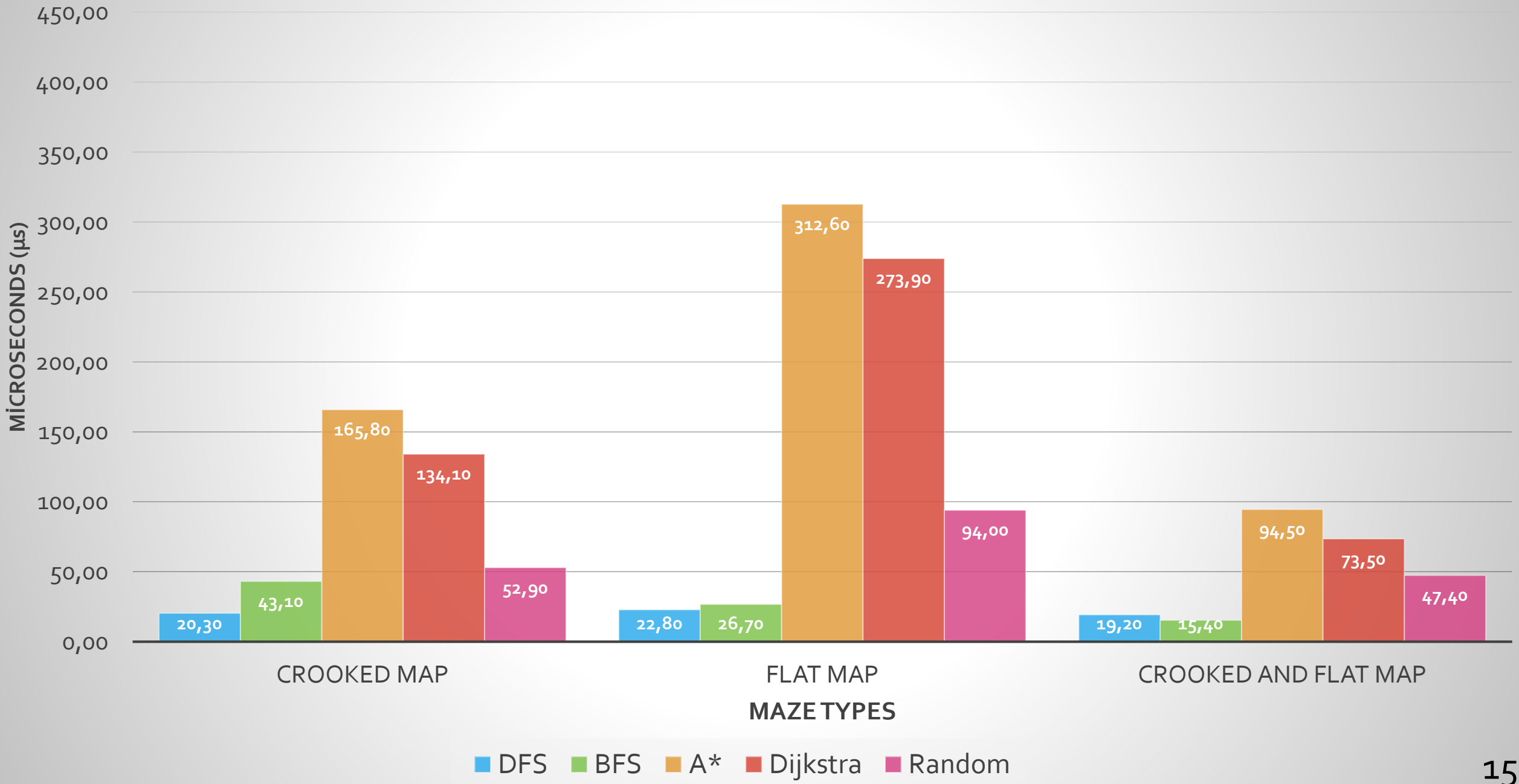
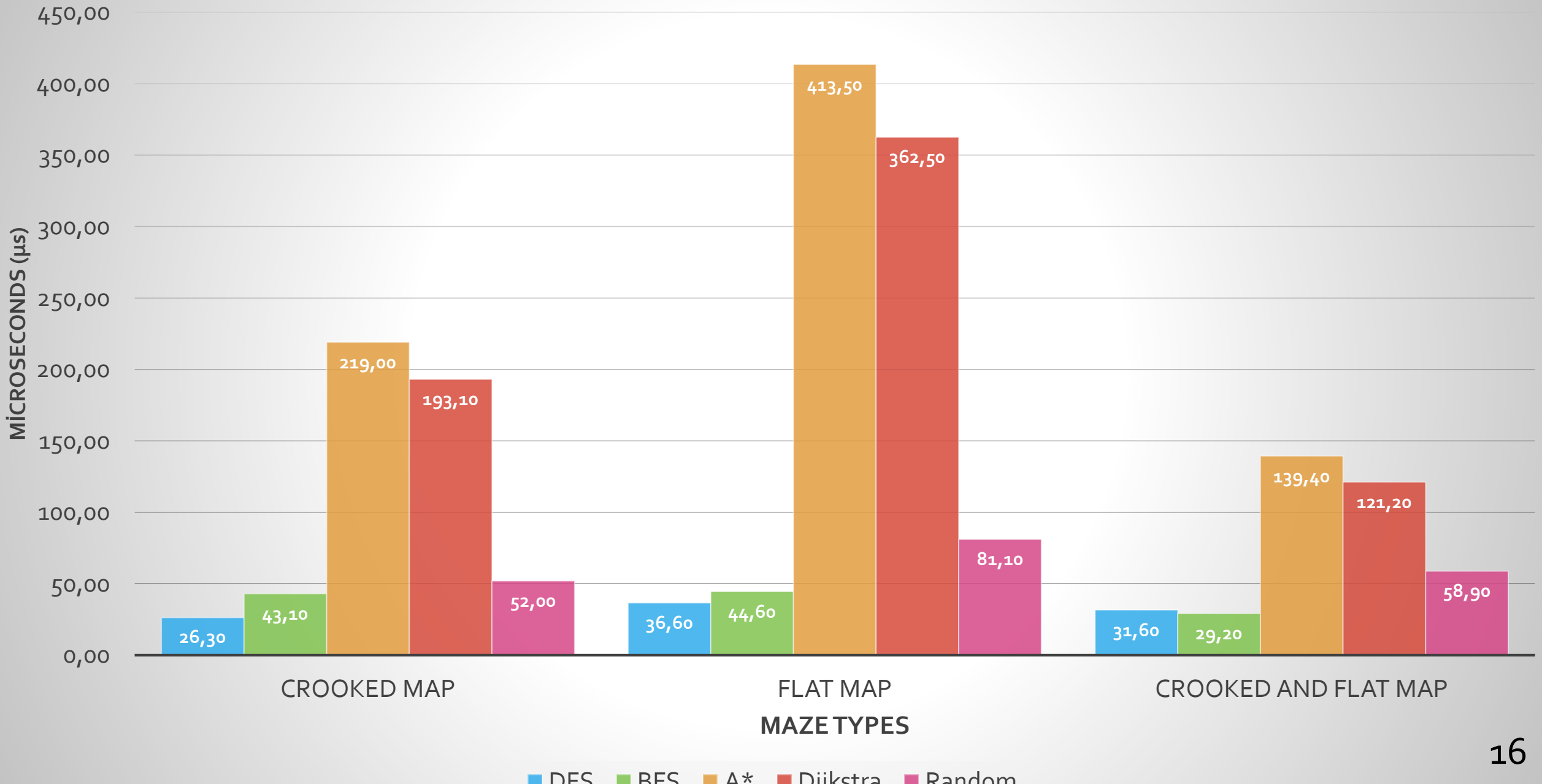


Figure 2-Total Run Time on PC2



Energy Consumption Graphics

- Graphics' data taken from experiments done on PC2. Experiment coded with C++. Computers' specifications given in the previous table.

Figure 3 - Energy Consumption of Crooked Maze

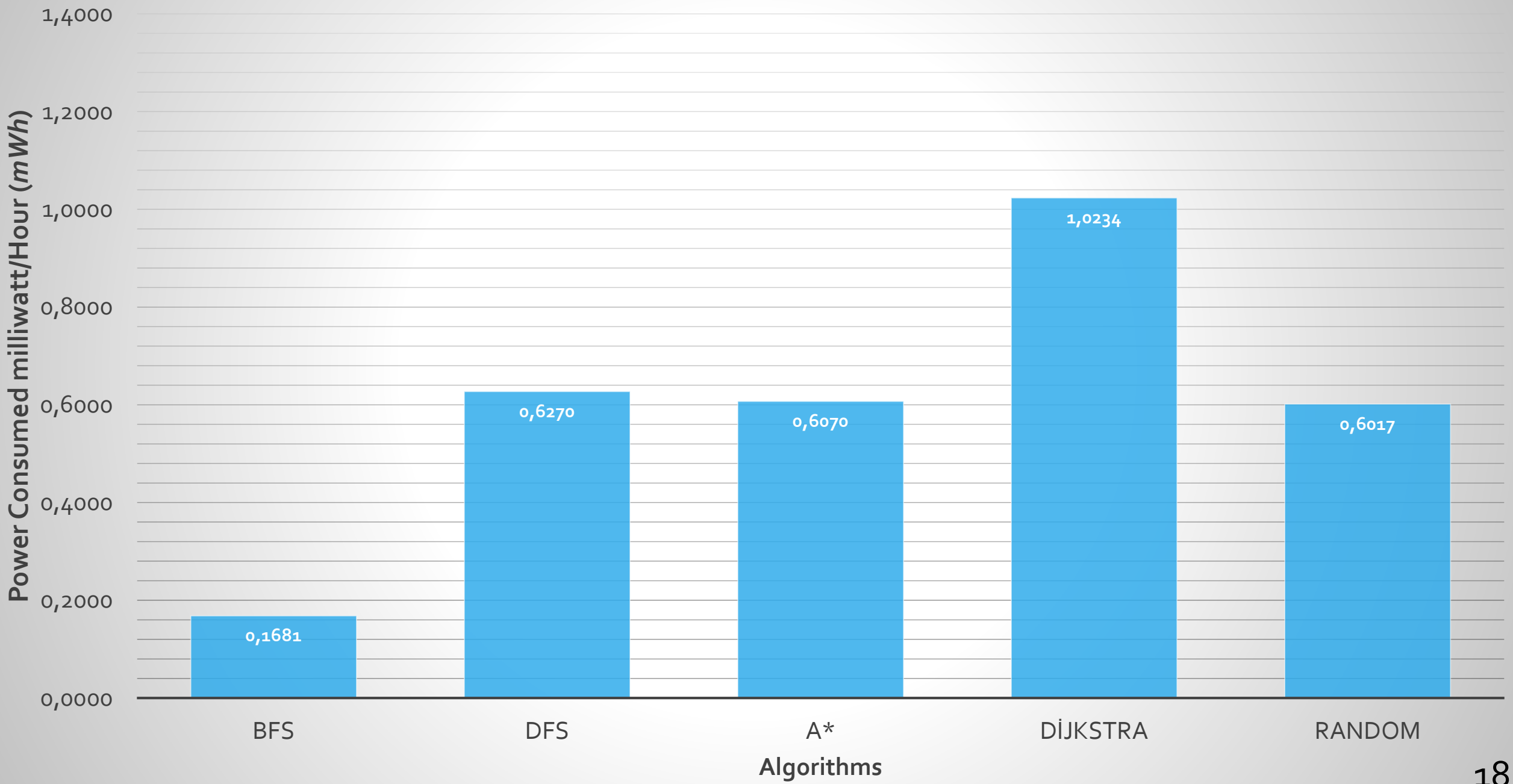


Figure 4 - Energy Consumption of Flat Maze

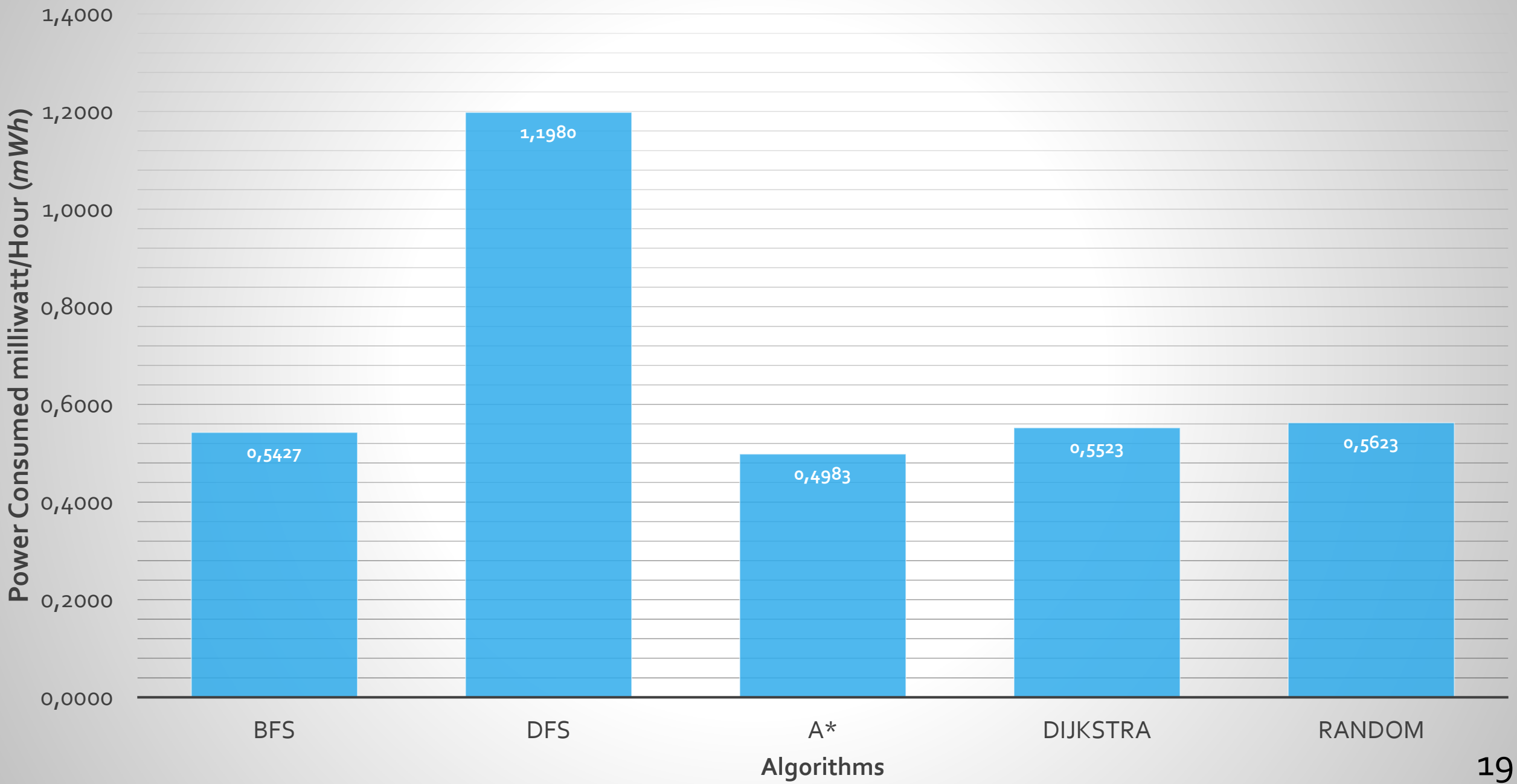
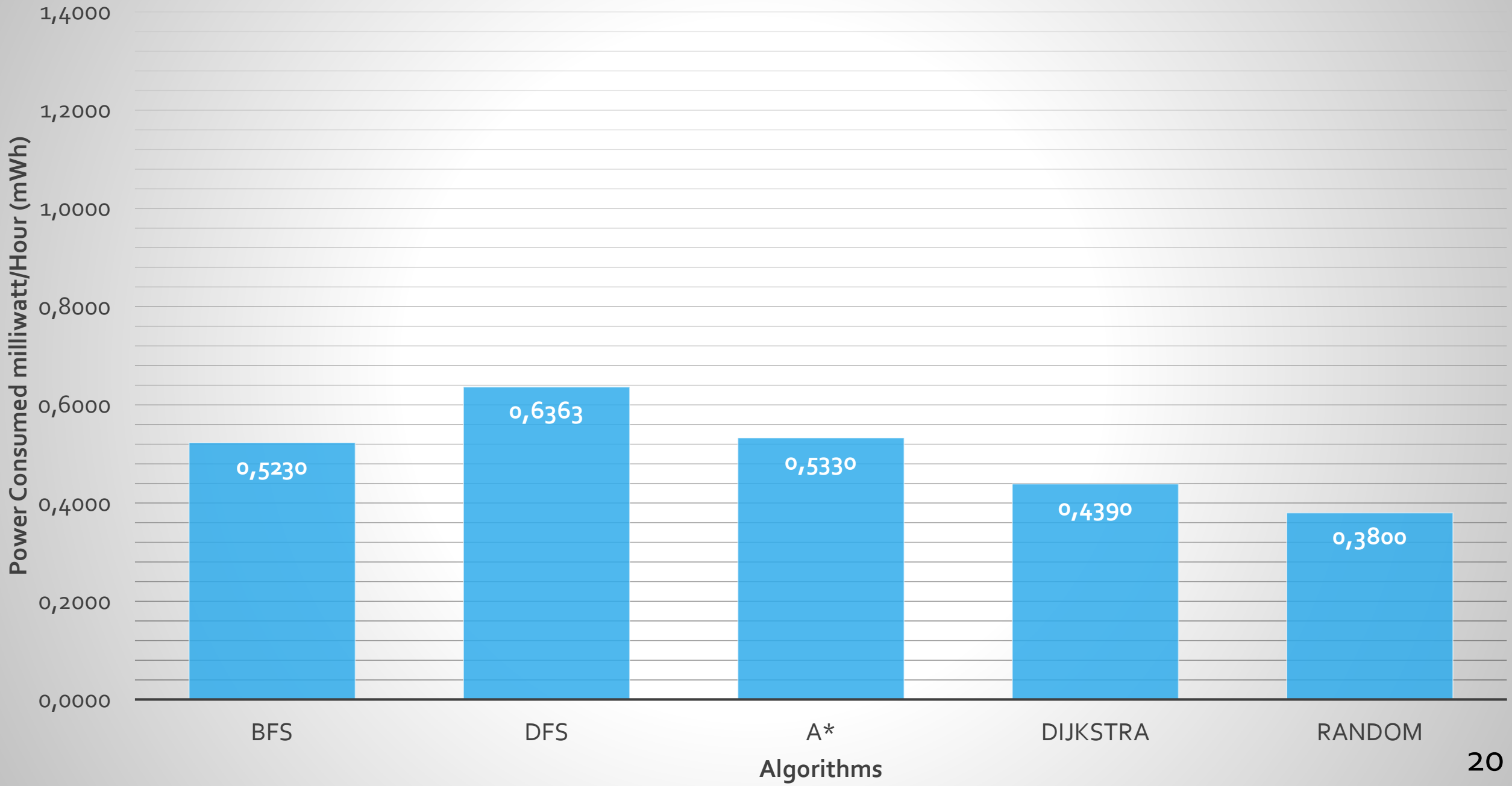


Figure 5 - Energy Consumption Flat & Crooked Maze



Conclusions

- In both computers A* and Dijkstra takes the most time to find a path.
- In both computers BFS and DFS takes the least time to find a path.
- All algorithms have a similar energy consumption rate on Flat & Crooked Maze.
- A* and Dijkstra have a big disadvantage on Flat Mazes. However, A* and Dijkstra's algorithms can be faster if implemented by trees created from the mazes. (Each pixel taken as a node in our implementation).
- Since all algorithms consume more time on Flat Mazes, an algorithm that transforms a maze to a tree might increase efficiency.
- Even though A* and Dijkstra takes the most time, they have a considerably fair amount of energy consumption.

Thank You For Your Time!