RBE 550 W6_Assignment 2

ANA* vs A* Introduction:

In this assignment, we have implemented Anytime Nonparametric A* algorithm and compared parameters like cost and time required to reach the goal node with the A* algorithm.

Pseudo Code:

ANA*:

```
ANA*()
15: G ← ∞; E ← ∞; OPEN ← ∅; ∀s : g(s) ← ∞; g(s_start) ← 0
16: Insert s_start into OPEN with key e(s_start)
17: while OPEN ≠ ∅ do
18:     ImproveSolution()
19:     Report current E-suboptimal solution
20:     Update keys e(s) in OPEN and prune if g(s) + h(s) ≥ G
```

```
ImproveSolution()
 1: while OPEN ≠ ∅ do
 2:     s ← arg max_{s∈OPEN} {e(s)}
 3:     OPEN ← OPEN \ {s}
 4:     if e(s) < E then
 5:         E ← e(s)
 6:     if IsGoal(s) then
 7:         G ← g(s)
 8:         return
 9:     for each successor s' of s do
10:         if g(s) + c(s,s') < g(s') then
11:             g(s') ← g(s) + c(s,s')
12:             pred(s') ← s
13:             if g(s') + h(s') < G then
14:                 Insert or update s' in OPEN with key e(s')
```

A*:

```
A* search {

closed list = [ ]
open list = [start node]

    do {
        if open list is empty then {
            return no solution
        }
        n = heuristic best node
        if n == final node then {
            return path from start to goal node
        }
        foreach direct available node do{
            if current node not in open and not in closed list do {
                add current node to open list and calculate heuristic
                set n as his parent node
            }
            else{
                check if path from star node to current node is
                better;
                if it is better calculate heuristics and transfer
                current node from closed list to open list
                set n as his parrent node
            }
        delete n from open list
        add n to closed list
    } while (open list is not empty)

}
```

Test Cases:

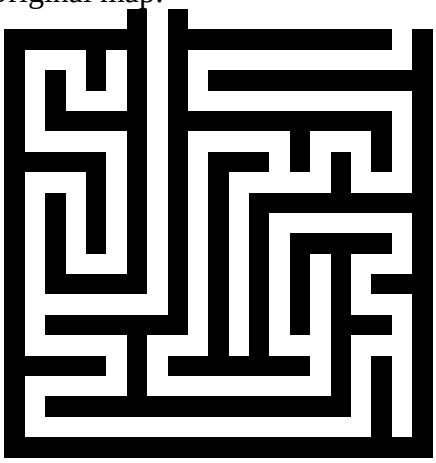
We have taken few test cases to compare the results of the ANA* algorithm with A*. The test cases are depicted with the original maps, and subsequently followed by the final results of both the algorithms. The test cases mentioned above are as follows:

- Trivial
- Medium
- Hard
- Very Hard

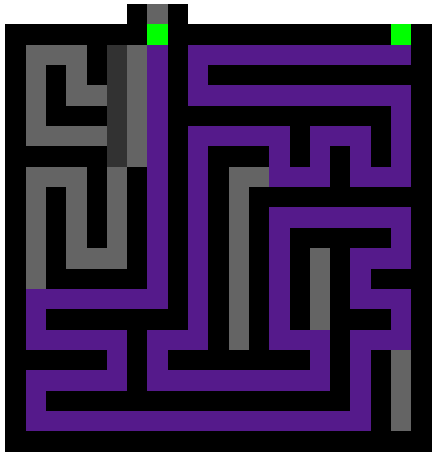
The difference between the above mentioned maps are their sizes. Increasing the size increases its complexity by many folds, and to test the robustness of ANA* and A*, we have implemented both the algorithms and obtained their output successfully.

TRIVIAL

Original map:

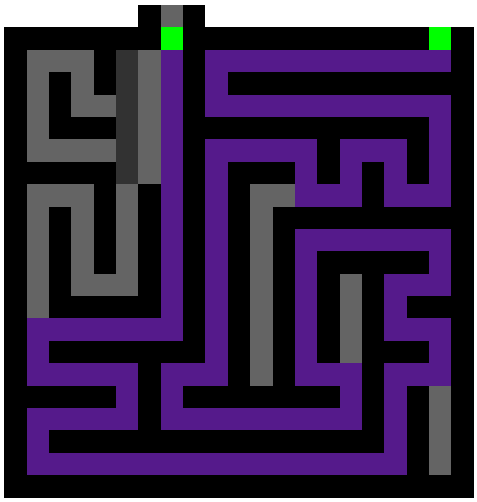


ANA* output:



ANA * Performance Data:
('Run-Time: ', 2.7298927307128906, ' milliseconds')
('Iterations: ', 1)
('G cost: ', 146)
('E cost: ', 38192.182646962625)

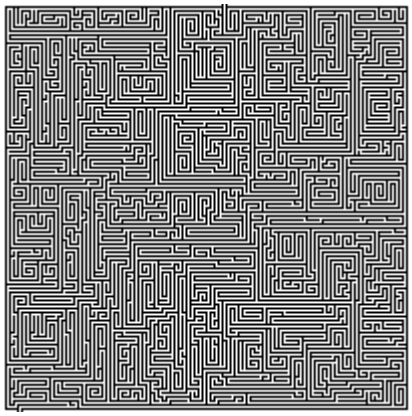
A* output:



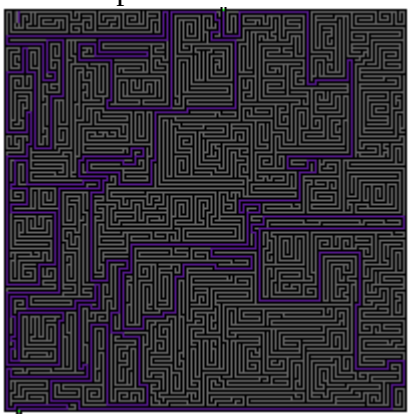
A* Performance Data :
('Run-Time: ', 2.758026123046875, 'milliseconds')
('Iterations: ', 1)
('G cost: ', 146)

MEDIUM

Original map:

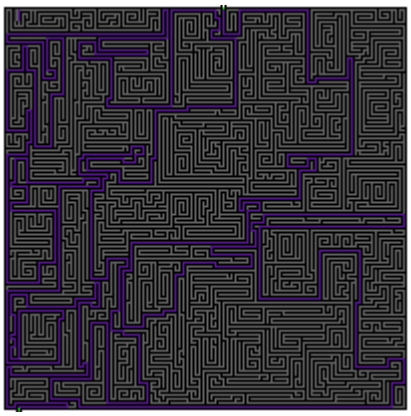


ANA* output:



ANA* Performance Data:
('Run-Time: ', 217.96703338623047, ' milliseconds')
('Iterations: ', 1)
('G cost: ', 3226)
('E cost: ', 4416.15913140638)

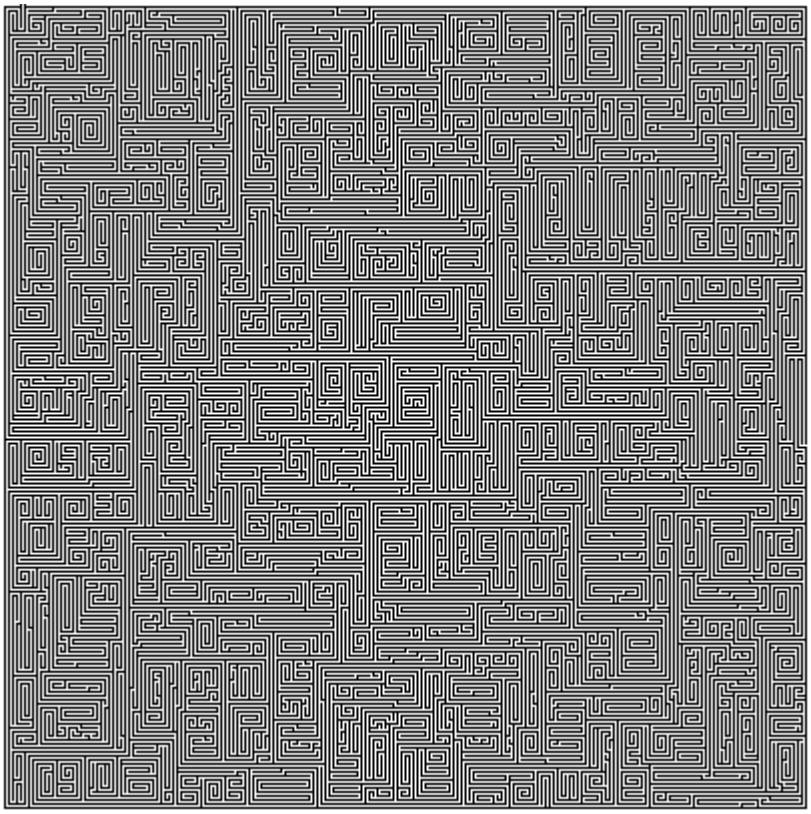
A* output:



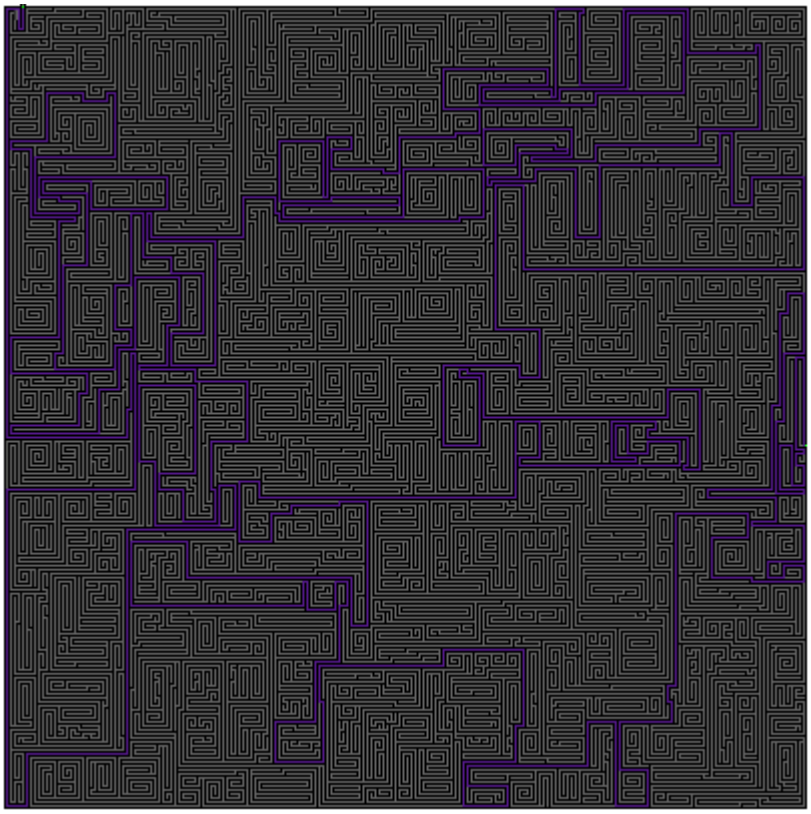
A* Performance Data:
('Run-Time: ', 0.032901763916015625, 'milliseconds')
('Iterations: ', 2)
('G cost: ', 3226)

HARD

Original map:

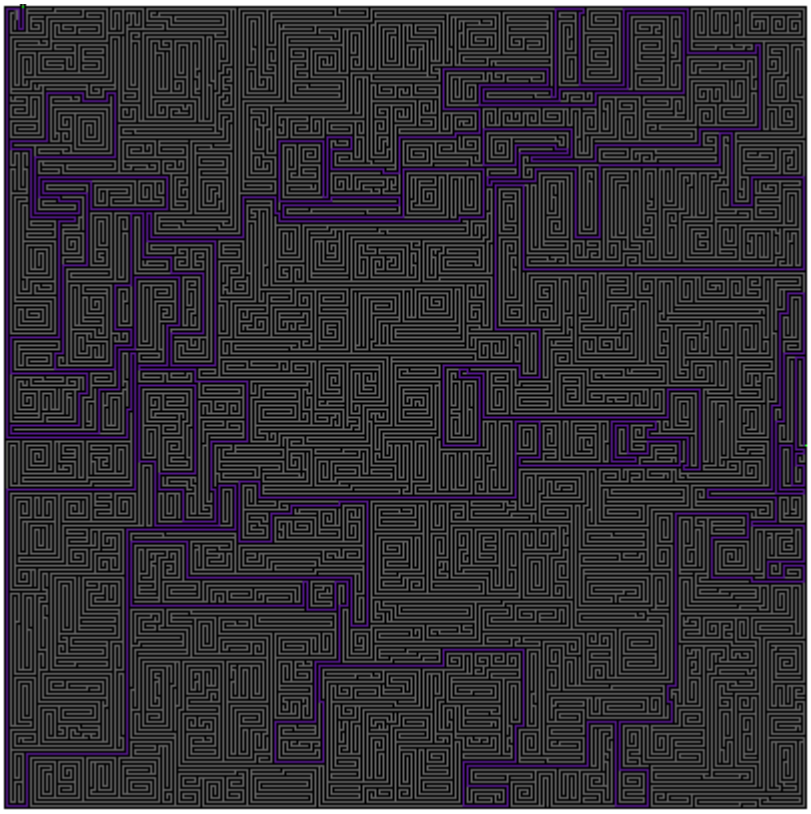


ANA* output:



ANA* Performance Data:
('Run-Time: ', 761.7208957672119, ' milliseconds')
('Iterations: ', 1)
('G cost: ', 8286)
('E cost: ', 2199.284195071731)

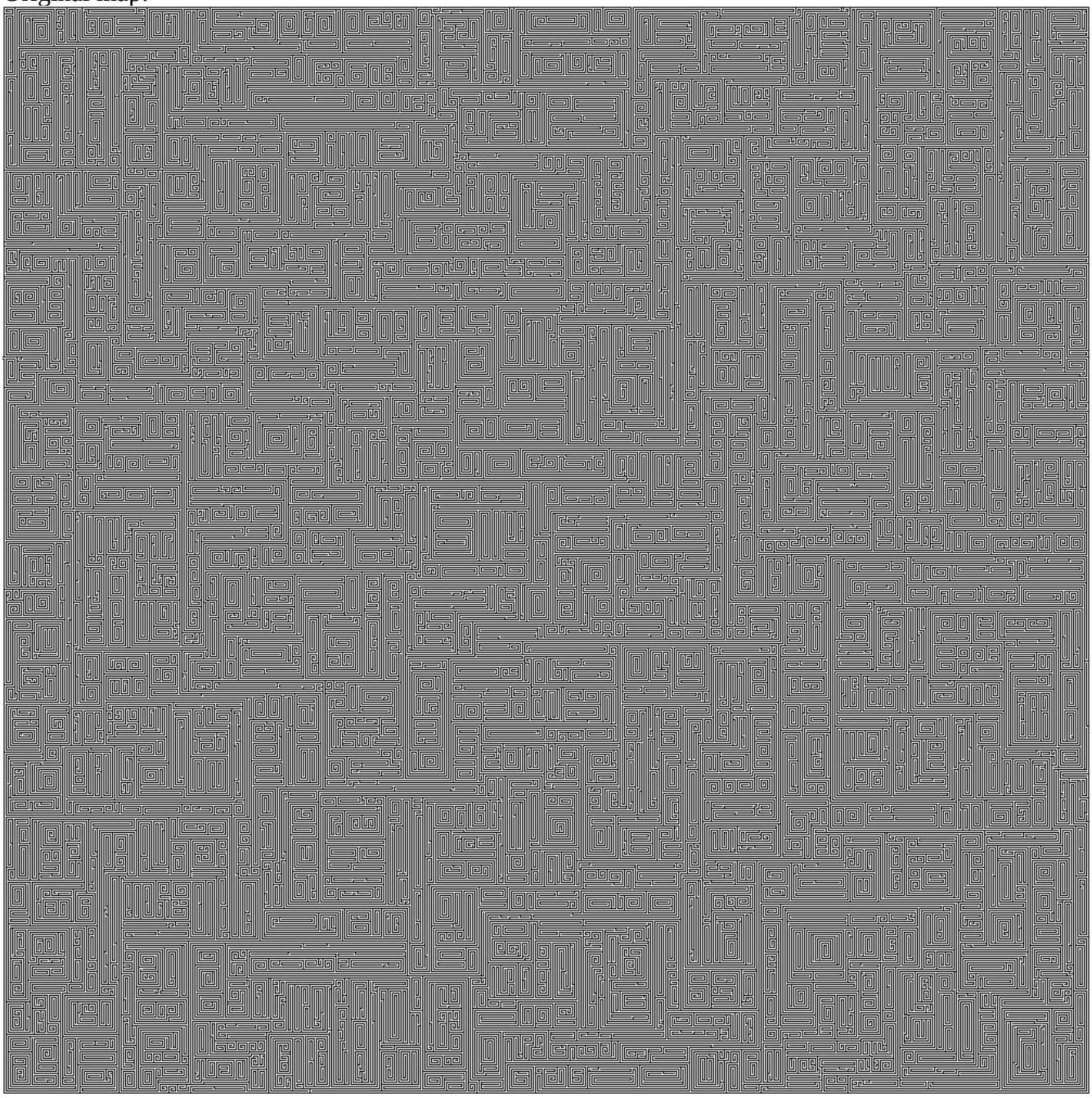
A* output:



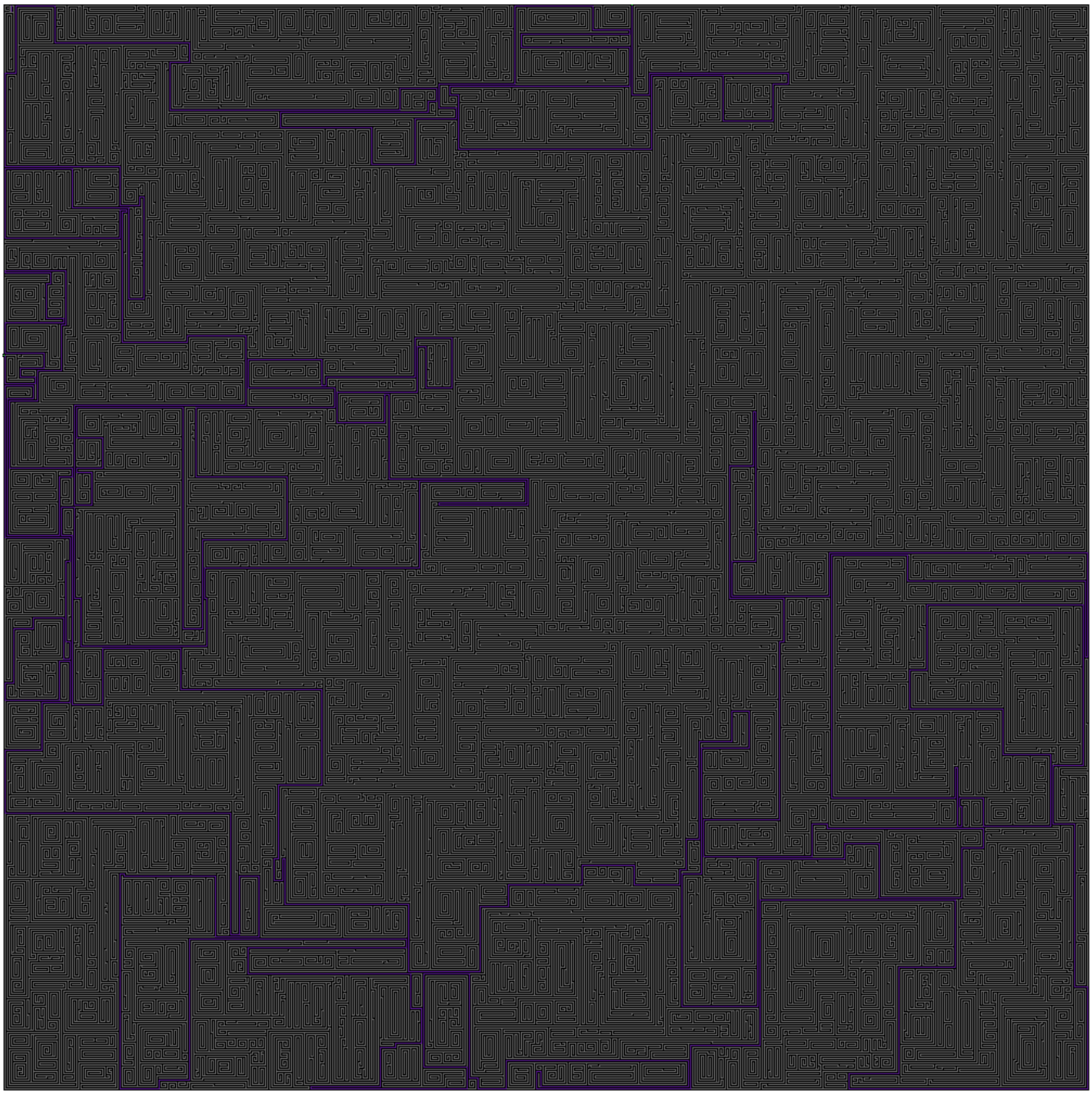
A* Performance Data:
('Run-Time: ', 162.9171371459961, 'milliseconds')
('Iterations: ', 2)
('G cost: ', 8286)

VERY HARD

Original map:

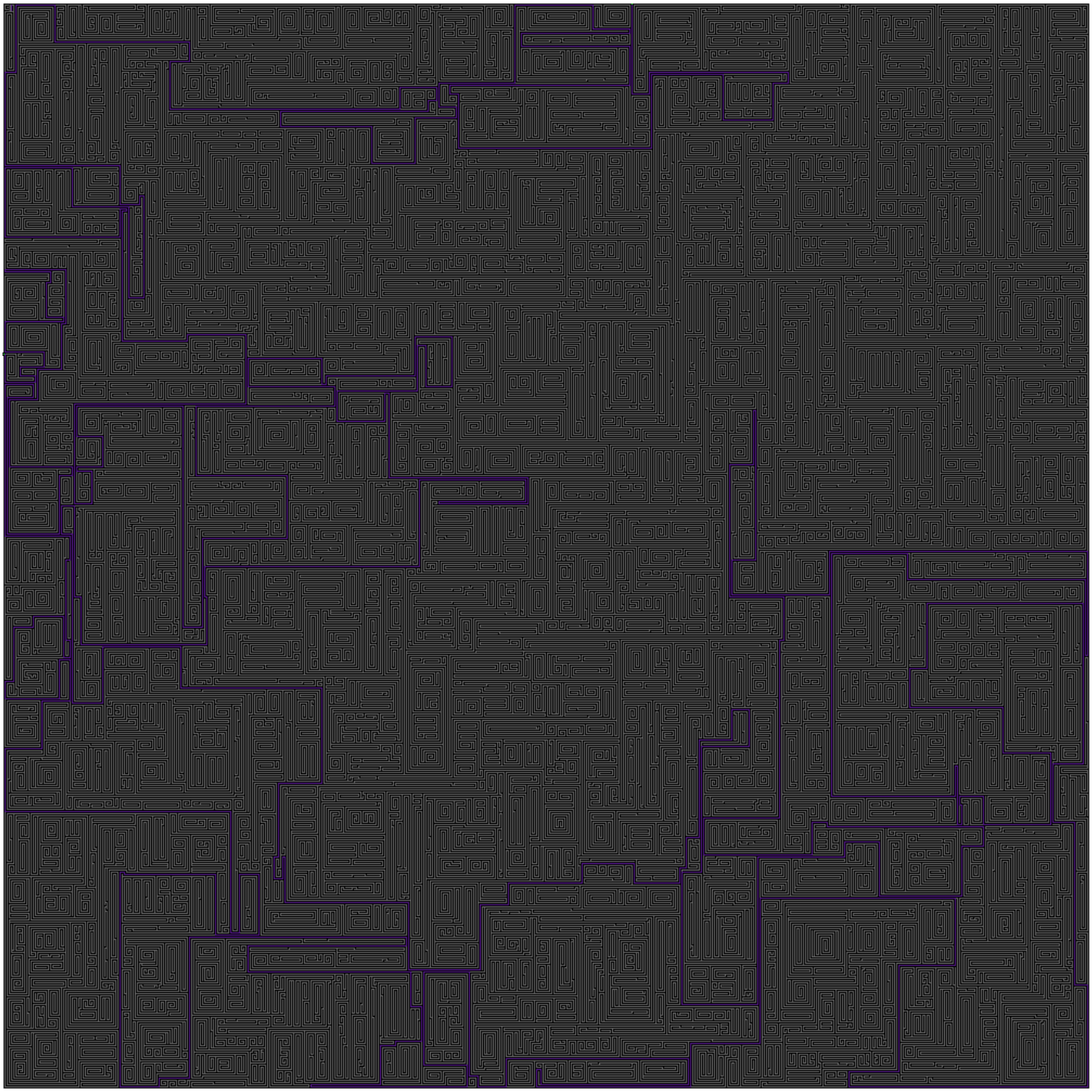


ANA* output:



ANA* Performance Data:
('Run-Time: ', 4901.829957962036, ' milliseconds')
('Iterations: ', 1)
('G cost: ', 18062)
('E cost: ', 864.6482879223693)

A* output:



A* Performance Data:
('Run-Time: ', 776.1249542236328, 'milliseconds')
('Iterations: ', 2)
('G cost: ', 18062)

Summary of performance:

Test cases	ANA*	A*
Trivial	• Run-Time: 2.729 ms • Iterations: 1 • G cost: 146 • E cost: 38192.1826	• Run-Time: 2.758 ms • Iterations: 1 • G cost: 146
Medium	• Run-Time: 217.967 ms • Iterations: 1 • G cost: 3226 • E cost: 4416.1591	• Run-Time: 0.0329 ms • Iterations: 2 • G cost: 3226
Hard	• Run-Time: 761.7208 ms • Iterations: 1 • G cost: 8286 • E cost: 2199.2841	• Run-Time: 162.917 ms • Iterations: 2 • G cost: 8286
Very Hard	• Run-Time: 4901.8299 ms • Iterations: 1 • G cost: 18062 • E cost: 864.6482	• Run-Time: 776.1249 ms • Iterations: 2 • G cost: 18062

Conclusion:

We were able to successfully implement ANA* and A* and analyze its output. The major difference between A* and ANA* was that ANA* calculates a maximal function e(s) which adaptively reduces to expand on the most promising node per iteration, and adapts to the greedy nature to improve its path quality. We found that ANA* took longer to find an optimal solution for all of the test cases than the A* search algorithm.