Assignment 3:

Bayesian Inference, Temporal State Estimation and Decision Making under Uncertainty

Team:

Eric Cajuste and Thurgood Kilper

Problem 1: Bayesian Networks

A. P(A,B,C,D,E) = P(A) P(B) P(C) P(D|A,B) P(E|B,C)
= (.2)(.5)(.8)(.1)(.3)
= 0.0024

B. P(~A,~B,~C,~D,~E) = P(~A)P(~B)P(~C)P(~D|~A,~B)P(~E|~B,~C)
= (.8)(.5)(.2)(.1)(.8)
= 0.0064

C. P(~A|B,C,D,E) = α P(~A) P(B) P(C) P(D|~A,B) P(E|B,C)
= α (.8)(.5)(.8)(.6)(.3)
= α 0.0576

P(A|B,C,D,E) = α P(A) P(B) P(C) P(D|A,B) P(E|B,C)
= α (.2)(.5)(.8)(.1)(.3)
= α 0.0024

1.0 = P(A|B,C,D,E) + P(~A|B,C,D,E)
= α 0.0024 + α 0.0576
= α (.06)

α = 16.66667

P(~A|B,C,D,E) = α 0.0576
= 0.96

Problem 2: Variable Elimination

A. P(Burglary(B)| JohnCalls(J) = T, MaryCalls(M) = T)
= α P(B=T, J=T, M=T)
= α P(B) Σ_E P(E) Σ_A P(J|A) P(A|E) P(M|A)
= α (.001) [(.002)((.7)(.95)(.9) + (.01)(.05)(.05)) + (.998)((.7)(.94)(.9) + (.01)(.06)(.05))]
= α (.001) [(.002)(.598525) + (.998)(.59223)]
= α (.001) [.00119705 + .59104554]
= α (.001) [.5922426]
= α 0.00059224

P(~Burglary(B)| JohnCalls(J) = T, MaryCalls(M) = T)

$= \alpha\, P(B=F, J=T, M=T)$
$= \alpha\, P(\sim B)\, \boldsymbol{\Sigma}_E\, P(E)\, \boldsymbol{\Sigma}_A\, P(J|A)\, P(A|E)\, P(M|A)$
$= \alpha\, (.999)\, [(.002)((.7)(.29)(.9) + (.01)(.71)(.05)) +$
$(.998)((.7)(.001)(.9) + (.01)(.999)(.05))]$
$= \alpha\, (.999)\, [(.002)(.183055) + (.998)(.0011295)]$
$= \alpha\, (.999)\, [.00036611 + .00112724]$
$= \alpha\, (.999)\, [.00149335]$
$= \alpha\, 0.00149186$

$1.0 = P(B|\ J, M) + P(\sim B|\ J, M)$
$= \alpha\, 0.00059224 + \alpha\, 0.00149186$
$= \alpha\, (0.00059224 + 0.00149186)$
$= \alpha\, (.0020841)$
$\alpha = 479.824$
$P(B|\ J, M) = \alpha\, 0.00059224$
$= .28417$

B. There is about 2 operations in our calculations compared to about 27 operations in tree enumeration.

C. The complexity for solving with enumeration would be $O(2^n)$ since each variable can take one of 2 values and directly depends on one other random variable, besides X_n.

The complexity for solving with variable elimination is O(n) since the work involved for solving the probability for any specific variable only involves at most one other variable. This would make the function for solving linear against time.

Problem 3: Markov Chain Monte Carlo

A. Markov Blanket Proof:

By the given diagram, the Markov Blanket of a variable X (MB(X)) consists of its parents, $(U_1, U_2 \ldots U_m)$, is children $(Y_1, Y_2 \ldots Y_i)$, and its children's parents $(Z_{1j} \ldots Z_{nj})$. The important property of a Markov Blanket states that a variable is independent from any other variables in its network given its Markov Blanket. By the rules of conditional independence, the computation of a joint distribution of cause and effect variables can be simplified using the Naive Bayesian model:

$$P(\text{Cause}, \text{Effect}_1, \ldots, \text{Effect}_n) = P(\text{Cause}) \prod_i P(\text{Effect}_i\, |\text{Cause})$$

As such, using the product rule, the probability of X using the Markov Blanket can be represented as:

$P(X \mid MB(X)) = \boldsymbol{\alpha}\, P(X, Y_1, ..., Y_n) = \boldsymbol{\alpha}\, P(X) \prod_i P(Y_i \mid \text{Cause})$

Since X is a conditional probability, it can be expanded to

$P(X \mid \text{parents}(X)) \Rightarrow \quad P(X \mid U_1, U_2 \dots U_m)$

Cause, which normally includes only X, is expanded to include all of the parents of Y:

$$\begin{aligned}\boldsymbol{\alpha}\, P(X, Y_1, ..., Y_n) &= \boldsymbol{\alpha}\, P(X \mid U_1, U_2 \dots U_m) \prod_i P(Y_i \mid (Z_{1j} \dots Z_{nj})) \\ &= P(X \mid MB(X))\end{aligned}$$

For any node B outside of MB(X), P(X | MB(X), B) = P(X | MB(X)).

B. MCMC State Space:

Given:

Query: P(rain|sprinkler=true, wetgrass=true)

Variables: <cloudy, sprinkler, rain, wetgrass>

At each step, MCMC will sample randomly from one of the non-evidence variables, rain and cloudy, taking into account its markov blanket. Sprinkler and wetgrass are fixed evidence variables, so MCMC only considers 4 possible states:

State 1: <true, true, true, true>
State 2: <false, true, true, true>
State 3: <false, true, false, true>
State 4: <true, true, false, true>

C. Transition Matrix:

Using the above states and the probabilities of the rain/sprinkler network, we may construct a transition matrix of size 4x4:

$$\begin{bmatrix} 0.416 & 0.04 & 0 & 0.198 \\ 0.05 & 0.124 & 0.792 & 0 \\ 0 & 0.198 & 0.46 & 0.2 \\ 0.72 & 0 & 0.01 & 0.095 \end{bmatrix}$$

Where:

P(1,1) = 0.5(0.5 * 0.8 * 0.1) + 0.5(0.8 * 0.99) =0.416
P(1,2) = 0.5 * 0.8 * 0.1 = 0.04

P(1,4) = 0.2 * 0.99	= 0.198
P(2,1) = 0.5 * 0.2 * 0.5	= 0.05
P(2,2) = 0.5(0.5 * 0.2 * 0.5) + 0.5(0.2 * 0.99)	=0.124
P(2,3) = 0.8 * 0.99	= 0.792
P(3,2) = 0.2 * 0.99	= 0.198
P(3,3) = 0.5(0.5 * 0.8 * 0.5) + 0.5(0.8 * 0.9)	= 0.46
P(3,4) = 0.5 * 0.8 * 0.5	= 0.2
P(4,1) = 0.8 * 0.9	= 0.72
P(4,3) = 0.5 * 0.1 * 0.2	= 0.01
P(4,4) = 0.5(0.5 * 0.1 * 0.2) + 0.5(0.2 * 0.9)	= 0.095

(transitions where 2 variables change are impossible)

Problem 4: Bayes Theorem

A. Expected Net gain:

Car Market Value: $4000. Without taking it to the mechanic:

With P=0.7 the car is bought in good condition at $3000, resulting in:

4000 - 3000 = $1000 profit.

With P=0.3 the car is bought in poor condition at $3000, needing an extra $1400 in repairs, resulting in:

4000 - 3000 - 1400 = -$400 loss.

Therefore the expected net gain is

(0.7)(1000) + (0.3)(-400) = 700 - 120 = $580 profit.

B. Applying Bayes Theorem:

Taking the mechanic's test into consideration:

With P=0.8 the car will pass the test if in good condition (q+)
With P=0.2 the car will fail the test if in good condition
With P=0.35 the car will pass the test if in bad condition (q-)
With P=0.65 the car will fail the test if in bad condition

Using Bayes Theorem, we calculate P(q+ | pass):

= [P(pass | q+) P(q+)] / P(pass)
= [P(pass | q+) P(q+)] / [P(pass | q+) P(q+)] + [P(pass | q-) P(q-)]
= [(0.8)(0.7)] / [(0.8)(0.7)] + [(0.35)(0.3)]
= 0.56 / (0.56+0.105)
= 0.8421 ~ 0.84

Similarly, we calculate P(q- | pass):

= [P(pass | q-) P(q-)] / [P(pass | q-) P(q-)] + [P(pass | q+) P(q+)]

= [(0.35)(0.3)] / [(0.35)(0.3)] + [(0.8)(0.7)]
= 0.1578 ~ 0.16

P(q+ | fail):
= [P(fail | q+) P(q+)] / [P(fail | q+) P(q+)] + [P(fail | q-) P(q-)]
= [(0.2)(0.7)] / [(0.2)(0.7)] + [(0.65)(0.3)]
= 0.4179 ~ 0.42

P(q- | fail):
= [P(fail | q-) P(q-)] / [P(fail | q-) P(q-)] + [P(fail | q+) P(q+)]
= [(0.65)(0.3)] / [(0.65)(0.3)] + [(0.2)(0.7)]
= 0.5820 ~ 0.58

Thus, if the car passes the test, there is a P=0.84 chance that the test was correct and the car is in good condition. If the car fails the test, there is a P=0.58 chance that the test is correct and the car is in poor condition.

C. Utility:

The utility in this case scales linearly with the value of the car, so we use the net gain calculation. The cost of the mechanic test is $100.

With P=0.84 the car passes the test in good condition, with value:
4000 - 3000 - 100 = 900
U = 900 * 0.84 = 756

With P=0.16 the car passes in bad condition, with value:
4000 - 3000 - 100 - 1400 = -500
U = -500 * 0.16 = -80
The expected utility of a passed car is 756-80= $676

With P=0.42 the car fails the test in good condition, with value:
4000 - 3000 - 100 = 900
U = 900 * 0.42 = 378

With P=0.58 the car fails in bad condition, with value:
4000 - 3000 - 100 - 1400 = -500
U = -500 * 0.58 = -290
The expected utility of a failed car is 378-290 = $88

In both cases of pass or fail, the car is expected to turn a profit, so the best decision is to buy the car.

D. Value of Information:

The expected value of the car without the test is $580.

With P=0.7 we judge the car to be good; taking the test in this case

increases the expected value to $676, an increase of $96
With P=0.3 we judge the car to be poor; taking the test in this case decreases the expected value to $88, a decrease of $492

The value of information is
(0.7)(96) + (0.3)(-492) = 67.2 - 147.6 = -$80.4

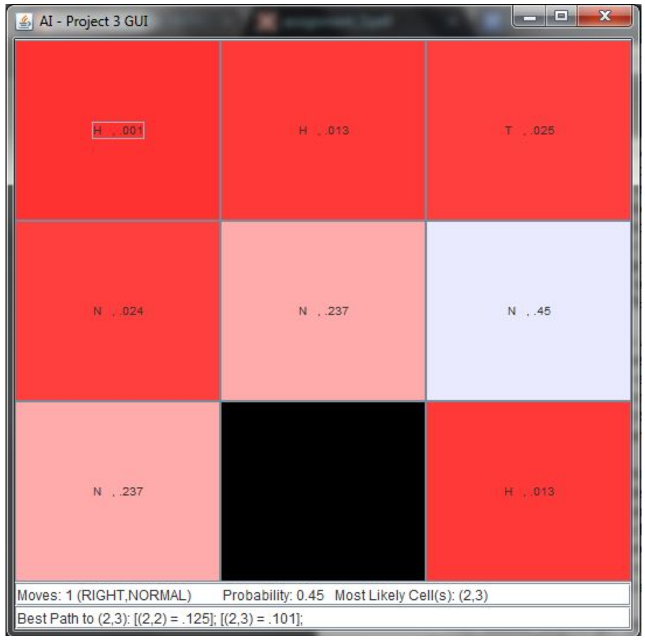
Taking the car to the mechanic is expected to depreciate its value, so we should choose not to take the test.

Problem 5: Programming Component
SMALL SCALE PROBLEM

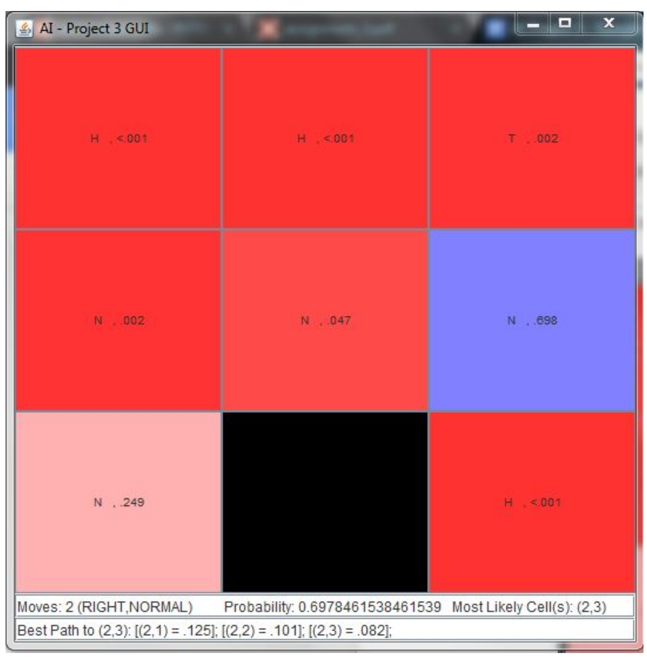
A. Filtering Problem:

After executing {right, right, down, down}, and receive {N,N,H,H}.

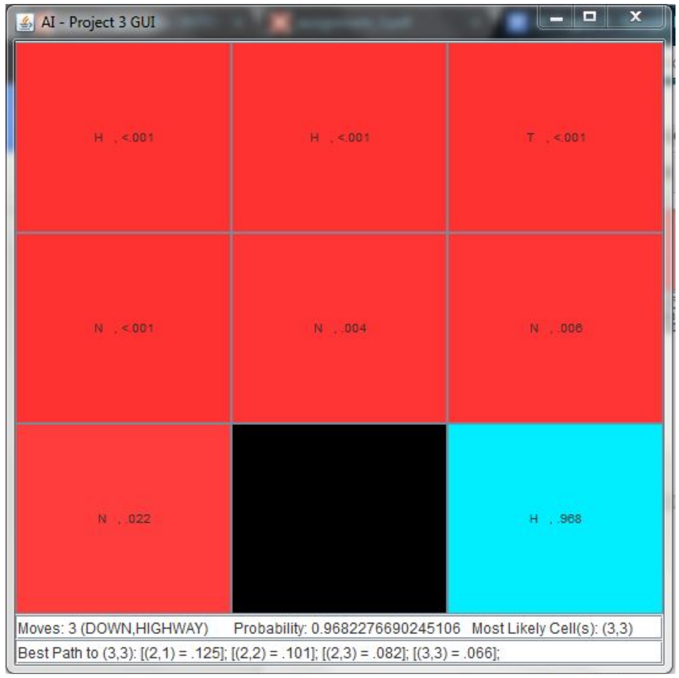
1. After 1st reading: Cell 2,3 was most probable.



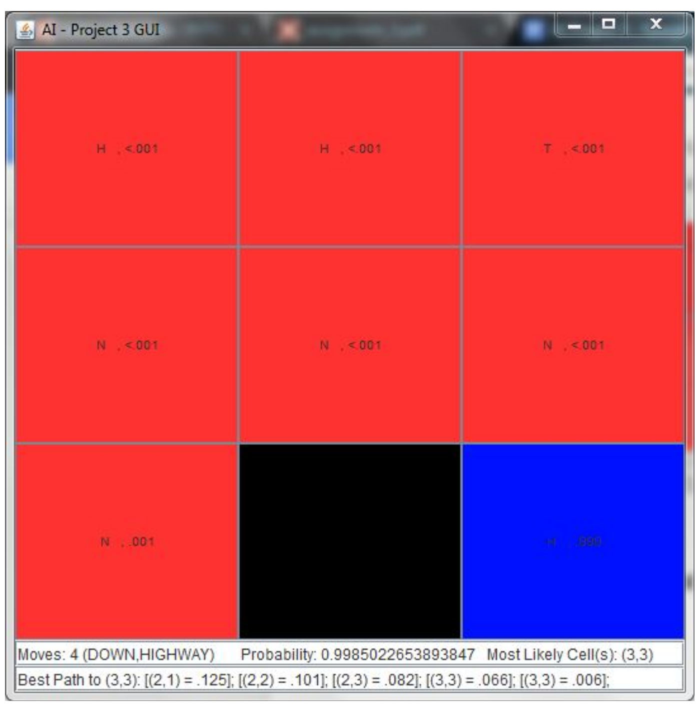
2. After 2nd reading: Cell 2,3 was most probable.



3. After 3rd reading: Cell 3,3 was most probable.



4. After 4th reading: Cell 2,3 was most probable.



B. Most Likely Sequence Ending at Cell:

1. (2,3) after 1st move = [(2,2) P = .125]; [(2,3) P = .101];
2. (2,3) after 2nd move = [(2,1) P = .125]; [(2,2) P = .101]; [(2,3) P = .082];
3. (3,3) after 3rd move = [(2,1) P = .125]; [(2,2) P = .101]; [(2,3) P = .082]; [(3,3) = .066];
4. (3,3) after 4th move = [(2,1) P = .125]; [(2,2) P = .101]; [(2,3) P = .082]; [(3,3) P = .066]; [(3,3) P = .006];

GENERATING GROUND TRUTH DATA

C. Example Ground Truth Paths (Refer to GTD files)

Example 1:

0: (8,37)		
1: (8,37)	D	N
2: (8,36)	L	T
3: (9,36)	D	N
4: (8,36)	U	H
5: (9,36)	D	N
6: (9,37)	R	N
7: (9,38)	R	H
8: (10,38)	D	N
9: (11,38)	D	H
10: (12,38)	D	T
11: (12,39)	R	T
12: (11,39)	U	T
13: (11,39)	R	T
14: (11,38)	L	H
15: (11,39)	R	T
16: (11,39)	D	T
17: (11,38)	L	N
18: (11,39)	R	T
19: (12,39)	D	T
20: (12,38)	L	T
21: (13,38)	D	H
22: (13,38)	L	H
23: (13,39)	R	T
24: (13,40)	R	N
25: (13,39)	L	H
26: (14,39)	D	N
27: (15,39)	D	N
28: (16,39)	D	T
29: (16,40)	R	T
30: (17,40)	D	H
31: (16,40)	U	T
32: (15,40)	U	H
33: (15,39)	L	N
34: (16,39)	D	T
35: (16,40)	R	T
36: (17,40)	D	H
37: (17,40)	L	H
38: (18,40)	D	N
39: (18,40)	R	N
40: (18,40)	U	T
41: (18,40)	U	N
42: (17,40)	U	H
43: (17,39)	L	N
44: (16,39)	U	T
45: (16,38)	L	N
46: (15,38)	U	N
47: (15,39)	R	N
48: (15,39)	R	N
49: (15,38)	L	N
50: (16,38)	D	N
51: (17,38)	D	N
52: (17,37)	L	N
53: (17,38)	R	N
54: (16,38)	U	N
55: (16,39)	R	T
56: (16,40)	R	T
57: (17,40)	D	H
58: (17,40)	U	H
59: (17,40)	R	H
60: (17,41)	R	N
61: (18,41)	D	N
62: (18,40)	L	N
63: (18,40)	D	N
64: (17,40)	U	H
65: (18,40)	D	N
66: (18,39)	L	N
67: (18,40)	R	N
68: (17,40)	U	H
69: (16,40)	U	T
70: (17,40)	D	H
71: (17,39)	L	N
72: (16,39)	U	T
73: (15,39)	U	N
74: (15,38)	L	N
75: (15,37)	L	N
76: (16,37)	D	T
77: (16,36)	L	T
78: (15,36)	U	N
79: (14,36)	U	T
80: (14,36)	U	T
81: (15,36)	D	N
82: (15,37)	R	N
83: (15,38)	R	N
84: (14,38)	U	N
85: (14,39)	R	N
86: (14,38)	L	N
87: (14,38)	R	N
88: (15,38)	D	N
89: (16,38)	D	N
90: (16,39)	R	T
91: (17,39)	D	N
92: (18,39)	D	N
93: (18,38)	L	N
94: (17,38)	U	T
95: (17,39)	R	N
96: (17,40)	R	H
97: (16,40)	U	N
98: (16,40)	R	T
99: (16,40)	R	T
100: (16,39)	L	T

Example 2:

0: (26,45)		
1: (26,46)	D	N
2: (25,46)	L	T
3: (25,47)	D	N
4: (25,47)	U	H
5: (25,48)	D	N
6: (25,49)	R	N
7: (25,48)	R	H
8: (25,48)	D	N
9: (25,48)	D	H
10: (25,47)	D	T
11: (25,46)	R	T
12: (25,45)	U	T
13: (25,45)	R	T
14: (25,45)	L	H
15: (26,45)	R	T
16: (25,45)	D	T
17: (26,45)	L	N
18: (25,45)	R	T
19: (25,45)	D	T
20: (24,45)	L	T
21: (24,46)	D	H
22: (24,46)	L	H
23: (24,46)	R	T
24: (25,46)	R	N
25: (25,45)	L	H
26: (24,45)	D	N
27: (24,45)	D	N
28: (24,46)	D	T
29: (25,46)	R	T
30: (26,46)	D	H
31: (27,46)	U	T
32: (28,46)	U	H
33: (27,46)	L	N
34: (27,45)	D	T
35: (27,44)	R	T
36: (27,43)	D	H
37: (27,42)	L	H
38: (26,42)	D	N
39: (27,42)	R	N
40: (26,42)	U	T
41: (26,41)	U	N
42: (25,41)	U	H
43: (25,42)	L	N
44: (25,41)	U	T
45: (25,40)	L	N
46: (24,40)	U	N
47: (23,40)	R	N
48: (23,41)	R	N
49: (24,41)	L	N
50: (23,41)	D	N
51: (23,42)	D	N
52: (23,43)	L	N
53: (23,44)	R	N
54: (23,44)	U	N
55: (23,45)	R	T
56: (24,45)	R	T
57: (24,45)	D	H
58: (25,45)	U	H
59: (24,45)	R	H
60: (25,45)	R	N
61: (25,46)	D	N
62: (24,46)	L	N
63: (24,46)	D	N
64: (24,45)	U	H
65: (25,45)	D	N
66: (26,45)	L	N
67: (25,45)	R	N
68: (25,46)	U	H
69: (25,45)	U	T
70: (25,45)	D	H
71: (24,45)	L	N
72: (24,45)	U	T
73: (24,46)	U	N
74: (24,46)	L	N
75: (24,46)	L	N
76: (23,46)	D	T
77: (22,46)	L	T
78: (23,46)	U	N
79: (23,46)	U	T
80: (23,45)	U	T
81: (23,46)	D	N
82: (24,46)	R	N
83: (25,46)	R	N
84: (26,46)	U	N
85: (26,45)	R	N
86: (26,46)	L	N
87: (26,47)	R	N
88: (25,47)	D	N
89: (25,48)	D	N
90: (25,47)	R	T
91: (25,47)	D	N
92: (25,46)	D	N
93: (25,45)	L	N
94: (26,45)	U	T
95: (26,46)	R	N
96: (27,46)	R	H
97: (27,45)	U	N
98: (28,45)	R	T
99: (29,45)	R	T
100: (30,45)	L	T

D. Sample Heat Maps

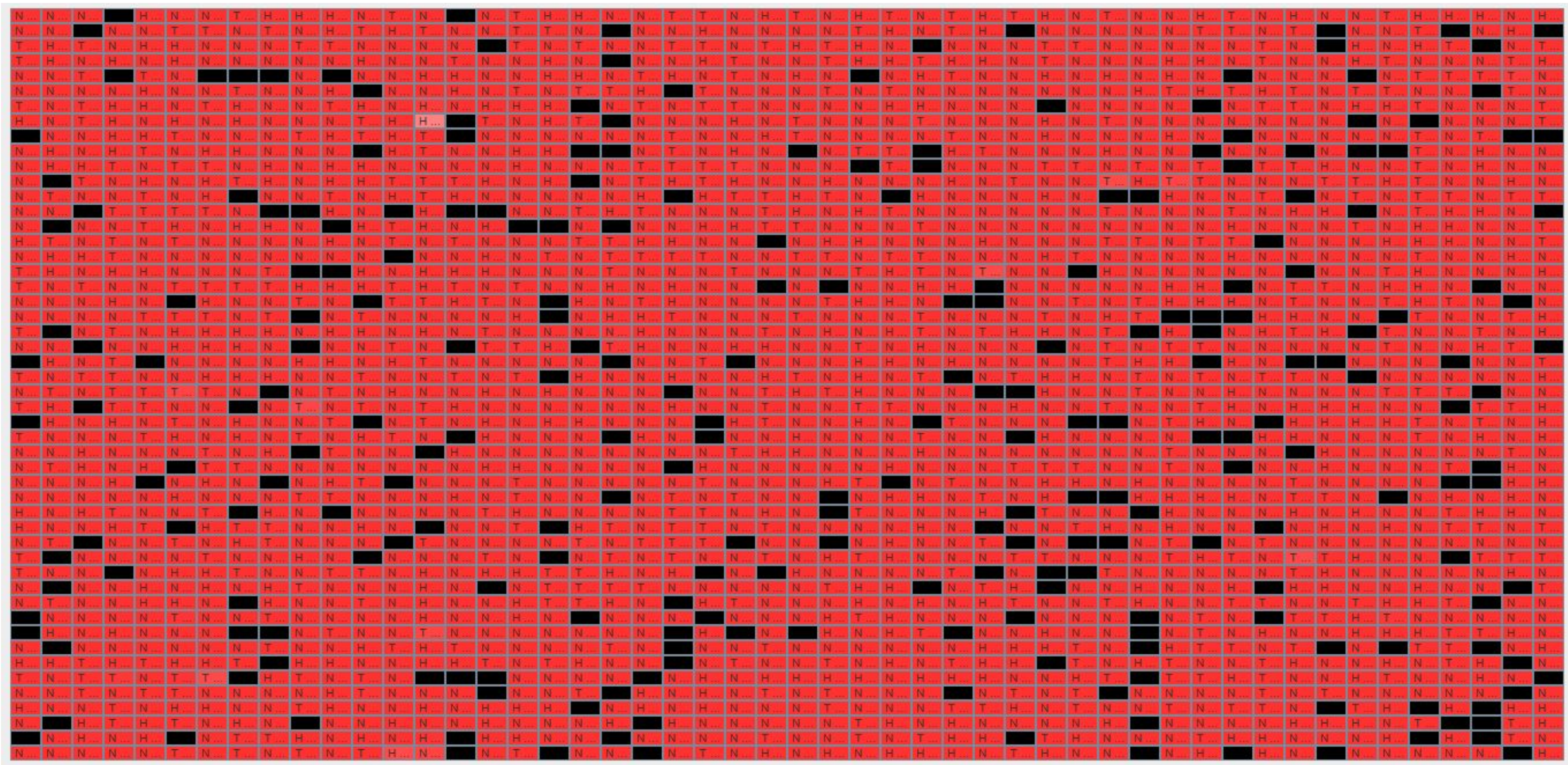
Note: We used a color gradient to visualize.

Dark red ~ very low probability (0% ~ 15%)

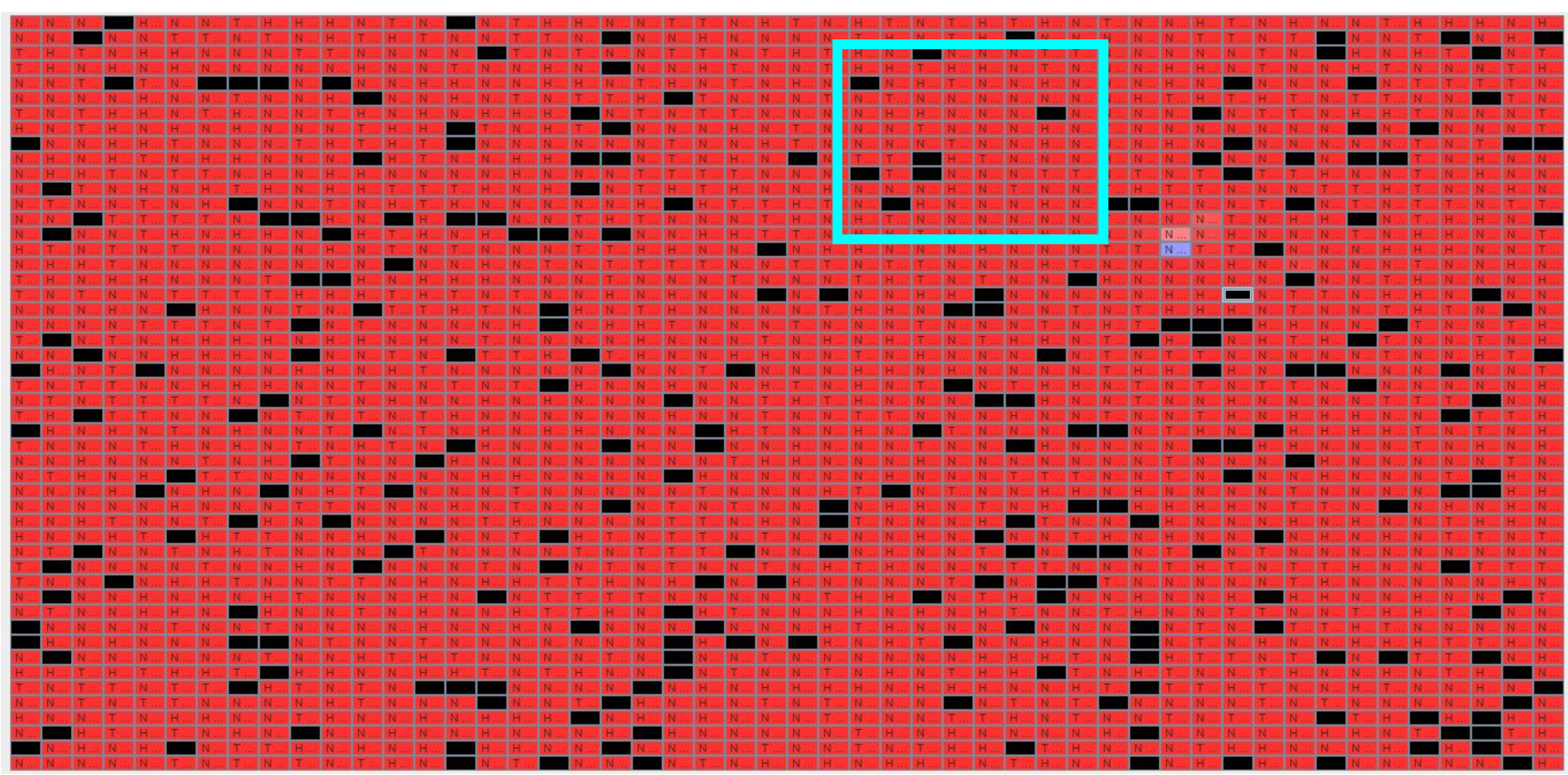
White ~ (~40%)

Dark Blue ~ highly probable (90% ~ 100%)

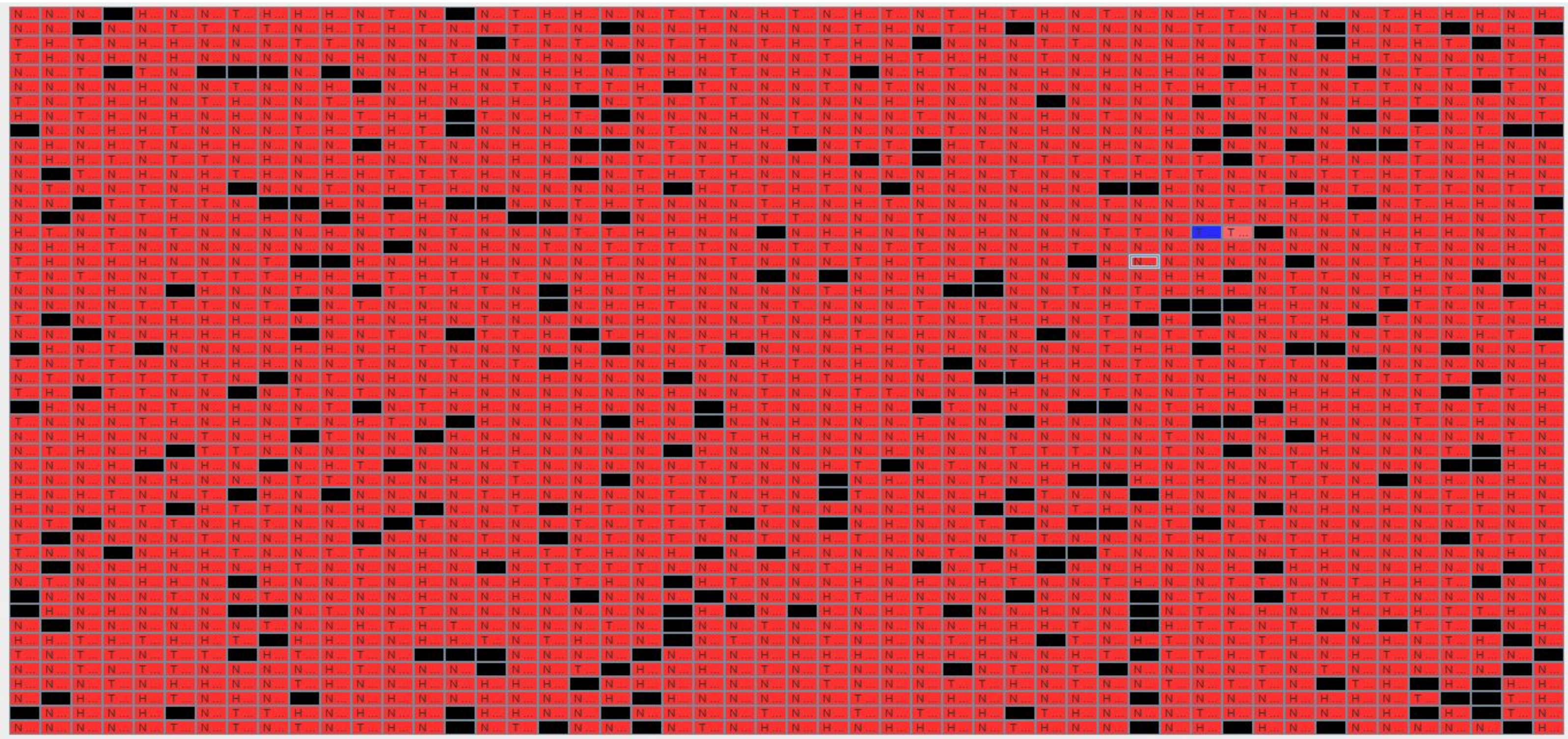
After 10 Moves: Cell 8,14 is most probable after 10 steps.



After 50 Moves: Cell 16,38 is most probable after 50 steps.



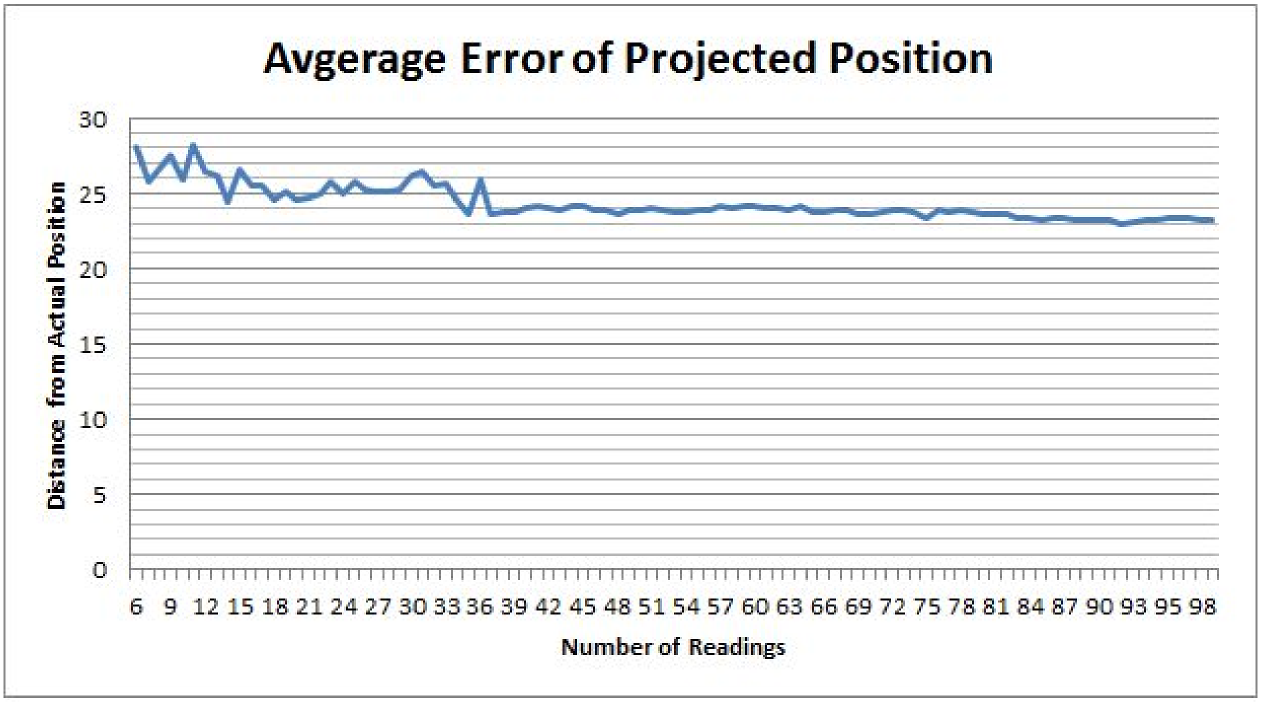
After 100 Moves: Cell 16,39 is most probable after 100 steps.

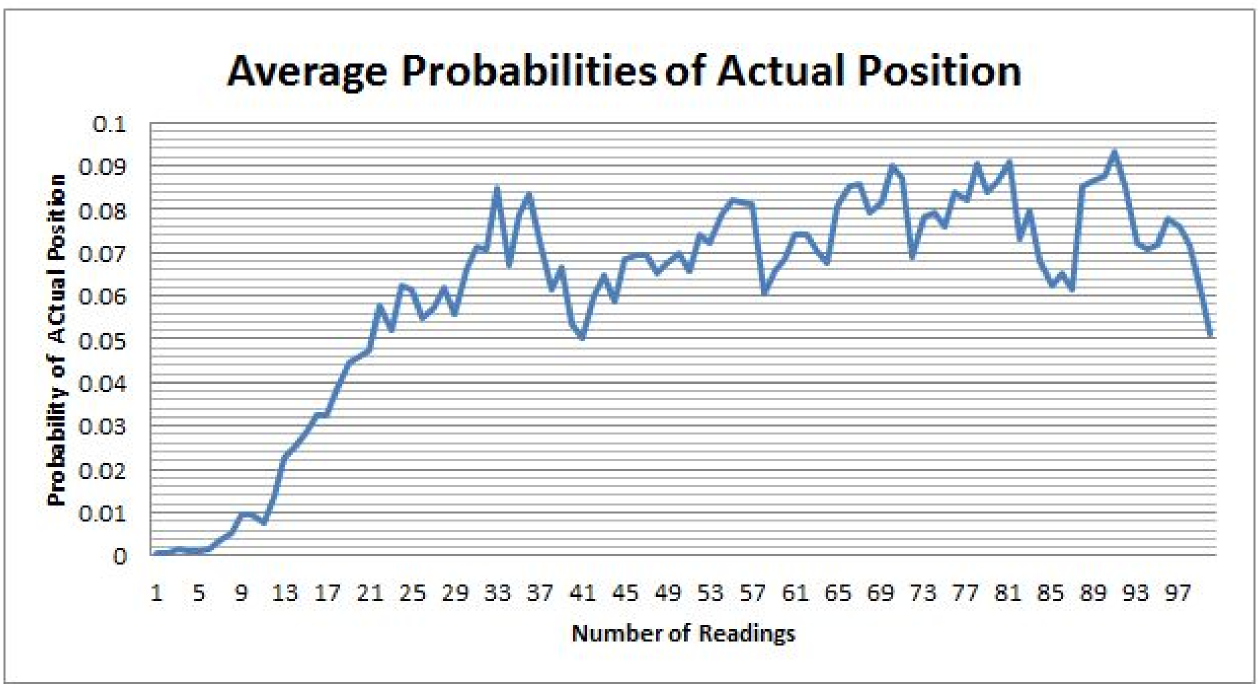


Path used for the 3 images:

0: (8,37)		
1: (8,37)	D	N
2: (8,36)	L	T
3: (9,36)	D	N
4: (8,36)	U	H
5: (9,36)	D	N
6: (9,37)	R	N
7: (9,38)	R	H
8: (10,38)	D	N
9: (11,38)	D	H
10: (12,38)	D	T
11: (12,39)	R	T
12: (11,39)	U	T
13: (11,39)	R	T
14: (11,38)	L	H
15: (11,39)	R	T
16: (11,39)	D	T
17: (11,38)	L	N
18: (11,39)	R	T
19: (12,39)	D	T
20: (12,38)	L	T
21: (13,38)	D	H
22: (13,38)	L	H
23: (13,39)	R	T
24: (13,40)	R	N
25: (13,39)	L	H
26: (14,39)	D	N
27: (15,39)	D	N
28: (16,39)	D	T
29: (16,40)	R	T
30: (17,40)	D	H
31: (16,40)	U	T
32: (15,40)	U	H
33: (15,39)	L	N
34: (16,39)	D	T
35: (16,40)	R	T
36: (17,40)	D	H
37: (17,40)	L	H
38: (18,40)	D	N
39: (18,40)	R	N
40: (18,40)	U	T
41: (18,40)	U	N
42: (17,40)	U	H
43: (17,39)	L	N
44: (16,39)	U	T
45: (16,38)	L	N
46: (15,38)	U	N
47: (15,39)	R	N
48: (15,39)	R	N
49: (15,38)	L	N
50: (16,38)	D	N
51: (17,38)	D	N
52: (17,37)	L	N
53: (17,38)	R	N
54: (16,38)	U	N
55: (16,39)	R	T
56: (16,40)	R	T
57: (17,40)	D	H
58: (17,40)	U	H
59: (17,40)	R	H
60: (17,41)	R	N
61: (18,41)	D	N
62: (18,40)	L	N
63: (18,40)	D	N
64: (17,40)	U	H
65: (18,40)	D	N
66: (18,39)	L	N
67: (18,40)	R	N
68: (17,40)	U	H
69: (16,40)	U	T
70: (17,40)	D	H
71: (17,39)	L	N
72: (16,39)	U	T
73: (15,39)	U	N
74: (15,38)	L	N
75: (15,37)	L	N
76: (16,37)	D	T
77: (16,36)	L	T
78: (15,36)	U	N
79: (14,36)	U	T
80: (14,36)	U	T
81: (15,36)	D	N
82: (15,37)	R	N
83: (15,38)	R	N
84: (14,38)	U	N
85: (14,39)	R	N
86: (14,38)	L	N
87: (14,38)	R	N
88: (15,38)	D	N
89: (16,38)	D	N
90: (16,39)	R	T
91: (17,39)	D	N
92: (18,39)	D	N
93: (18,38)	L	N
94: (17,38)	U	T
95: (17,39)	R	N
96: (17,40)	R	H
97: (16,40)	U	N
98: (16,40)	R	T
99: (16,40)	R	T
100: (16,39)	L	T

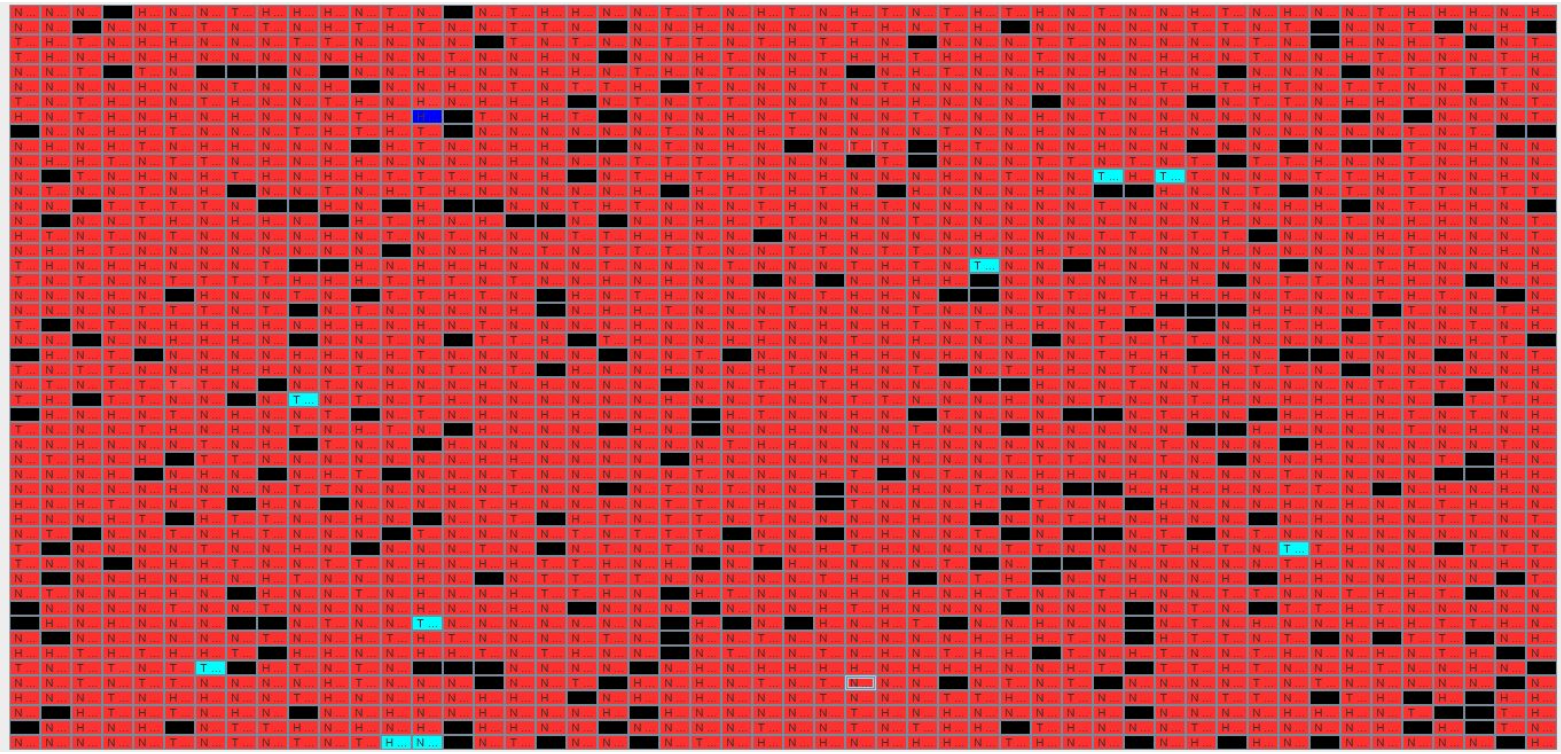
E. Best Projected Position Error and Actual Position Probabilities (Refer also to All-Probabilities.txt and All-Error.txt)



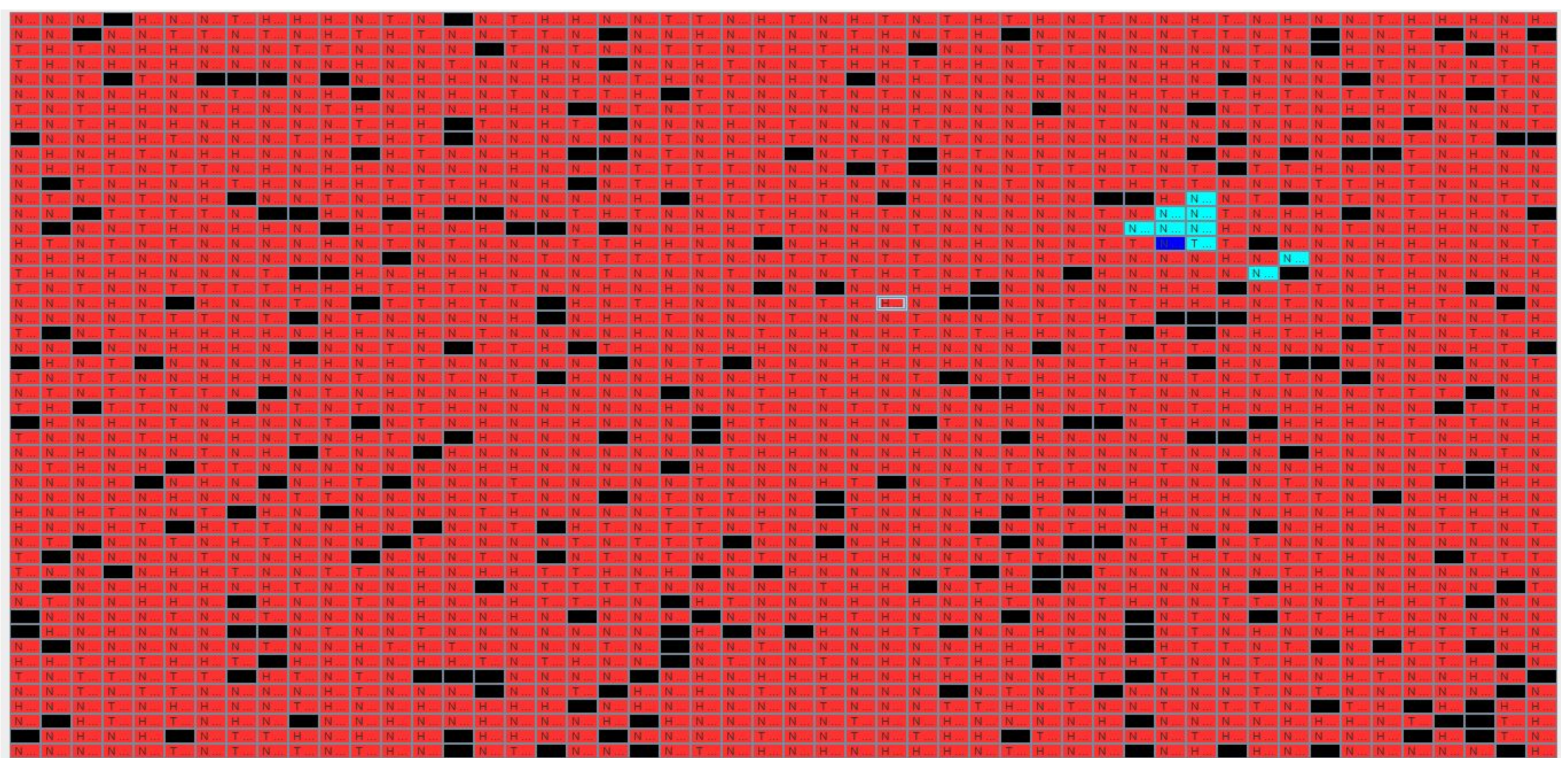


F. Top 10 Positions Examples (Note: Blue cell is most probable cell)

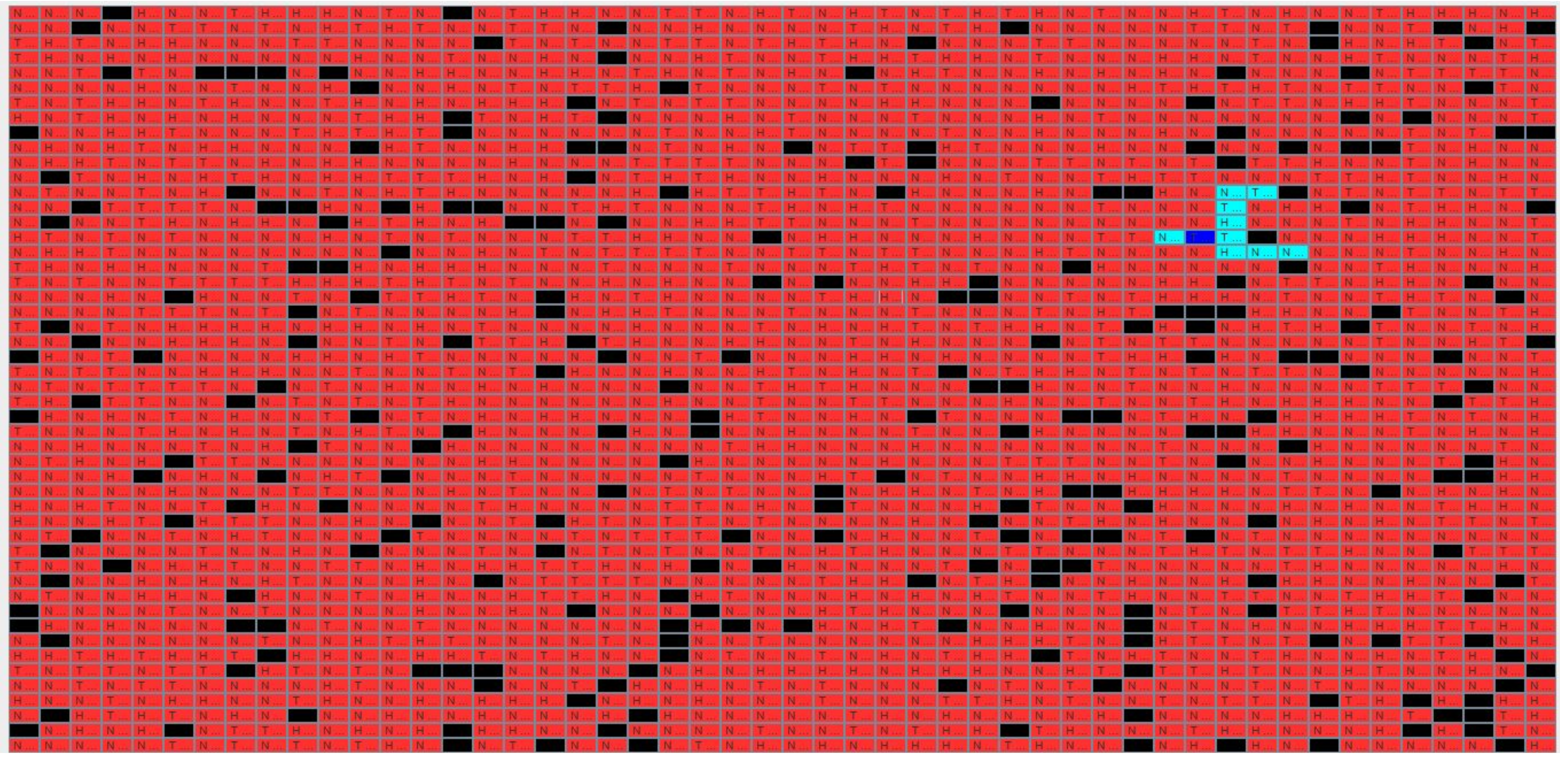
After 10 Moves:



After 50 Moves:



After 100 Moves:



G. Best Path Error

H. Computational Approximations

The biggest feature we have in our implementation is the addition of an array for every cell in the grid that contains arraylists depicting the best path to that particular cell. The array index of the array list would correspond to the length of the path from beginning to the cell.

The Viterbi algorithm would use the given input and determine the best path to a cell, but the process is recursive and includes many repeated calculations in order to complete. To reduce the number of calculations, we introduced an array in every cell that would keep track of all the paths and refer back to the array when the best path to a cell of a certain length was already calculated.

Before introducing this data structure, our program struggled to calculate viterbi sequences in a reasonable amount of time (minutes). But with the new structure, it is able to calculate the best path for the entire ground truth sequence in just a few seconds.