${\bf Homework}~{\bf 5}$

 ${\bf MacMillan,\ Kyle}$

November 26, 2018

${\bf Contents}$

Title

Table of	Contents
----------	----------

Li	st of	Figure	es																		i
1		apter 1 Proble	1 em 3		 	 	 •	 						 •					•	•	1 1
2		apter 1																			2
	2.1	Proble	em 1		 	 		 													2
	2.2	Proble	em 2		 			 													2
3	Cha	apter 1	7																		3
	3.1	Proble	em 2		 	 	 	 													3
	3.2	Proble	em 3		 	 		 													3
	3.3	Proble	em 4		 	 		 													3
		3.3.1	Problem																		3
		3.3.2	Problem																		3
		3.3.3	Problem	4c	 	 		 													3
		3.3.4	Problem																		3
		3.3.5	Problem																		3
4	Cha	apter 1	8																		4
			m 1		 	 		 													4
		4.1.1	Problem																		
		4.1.2	Problem																		
		4.1.3	Problem																		

List of Figures

1	WaveFront Distance Evaluation	1
2	WaveFront Shortest Path	1

1.1 Problem 3

To accomplish this problem I created a class to generate a random map of any size you want up to 99. It generates a random number of obstacles of random size. Play around with it, it's fun. You can change the random numbers or just run the file multiple times. Code for this problem can be found here.

Demonstration is shown in Figure 1. This is an application of the WaveFront BFS algorithm. From there it goes on to find the shortest path as seen in Figure 2.

The seed for that particular example is: 7635686187880284248

SS 01 02 03 04	05 16 15	16 17 18 19 20
	06 14	
02 03 04 05 06	07 08 12 13	14 15 16 17 18
03 04 05 06 07	08 09 10 11 12	13 14 15 16 17
04 05 06 07 08	09 10 11 12 13	14 15 16 17 18
	10 11 12 13 14	
06 07 08 09 10	11 12 13 14 15	16 17 18 19 20
07 08 09 10 11	12 13 <u>14</u> 15 16	<u>17</u> 18 19 20 21
08 09 10 11 12	13 14 16 17	19 20 21 22
09 10 11 12 13	14 15 17 18	20 21 22 23
10 11 12 13 14	15 16 18 19	21 22 23 24
11 <u>12</u> 13 <u>14</u> 15		
	17 <u>21</u> 20 21	
13 <u>15</u> <u>17</u>	18 <u>19</u> <u>21</u> 22	26 26 27
14	19 23	24 25 <u> </u> 27 GG

Figure 1: WaveFront Distance Evaluation



Figure 2: WaveFront Shortest Path

The book constantly refers to this as a start to goal process but that is not how flood-fill works because you can not guarantee shortest path; you can not have it both ways. The book calls for "minimal path" but then uses a non-optimal solver. I wrote it as the book asked the first time around then changed it to actually give the optimal answer because that's what we want.

To clarify: Starting from the start is fine but you have to traverse it backwards from goal to start. The reason is because you do not know which of the "equal" numbers to choose when you're walking from start to goal. You **know** which path is the shortest if you walk from goal to start. If two numbers are the same then either path is optimal/minimal.

2.1 Problem 1

asdf

2.2 Problem 2

Given 40 measurements at 2 meters we can subtract the 2 meters and then take the mean:

```
A 0.23521735B -0.13556097C 0.33806001
```

That is how far off each senor averages from 2 meters. Each new sensor reading then has that mean subtracted from it to yield:

```
A 2.22255225B 2.03233526C 1.79719609
```

We can then take the mean of those values for an expected distance of: 2.0173612016666667 meters. The code for this can be found here and in the code snippet 2.2.

```
dist_sens -= 2
mean = np.mean(dist_sens, dtype=np.float64, axis=0)
new_sens = np.array([[2.4577696, 1.8967743, 2.1352561]]) - mean
distance = np.mean(new_sens)
```

3.1 Problem 2

asdf

3.2 Problem 3

 asdf

3.3 Problem 4

3.3.1 Problem 4a

asdf

3.3.2 Problem 4b

 asdf

3.3.3 Problem 4c

 asdf

3.3.4 Problem 4d

 asdf

3.3.5 Problem 4e

 asdf

4.1 Problem 1

4.1.1 Problem 1.1

 asdf

4.1.2 Problem 1.2

 asdf

4.1.3 Problem 1.3

 asdf