KiDOBOTiKZ™
Engineering Real Engineers

Flood Fill Algorithm

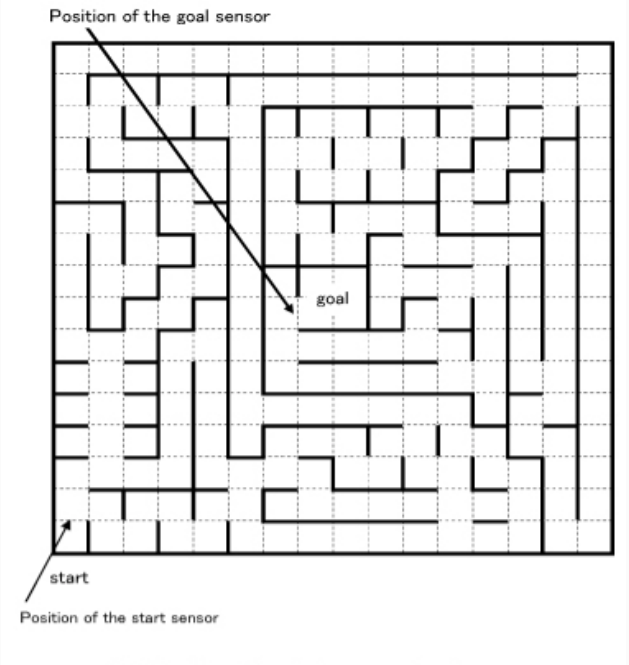
Contributions by Vishal Desh, Balasubramanian.

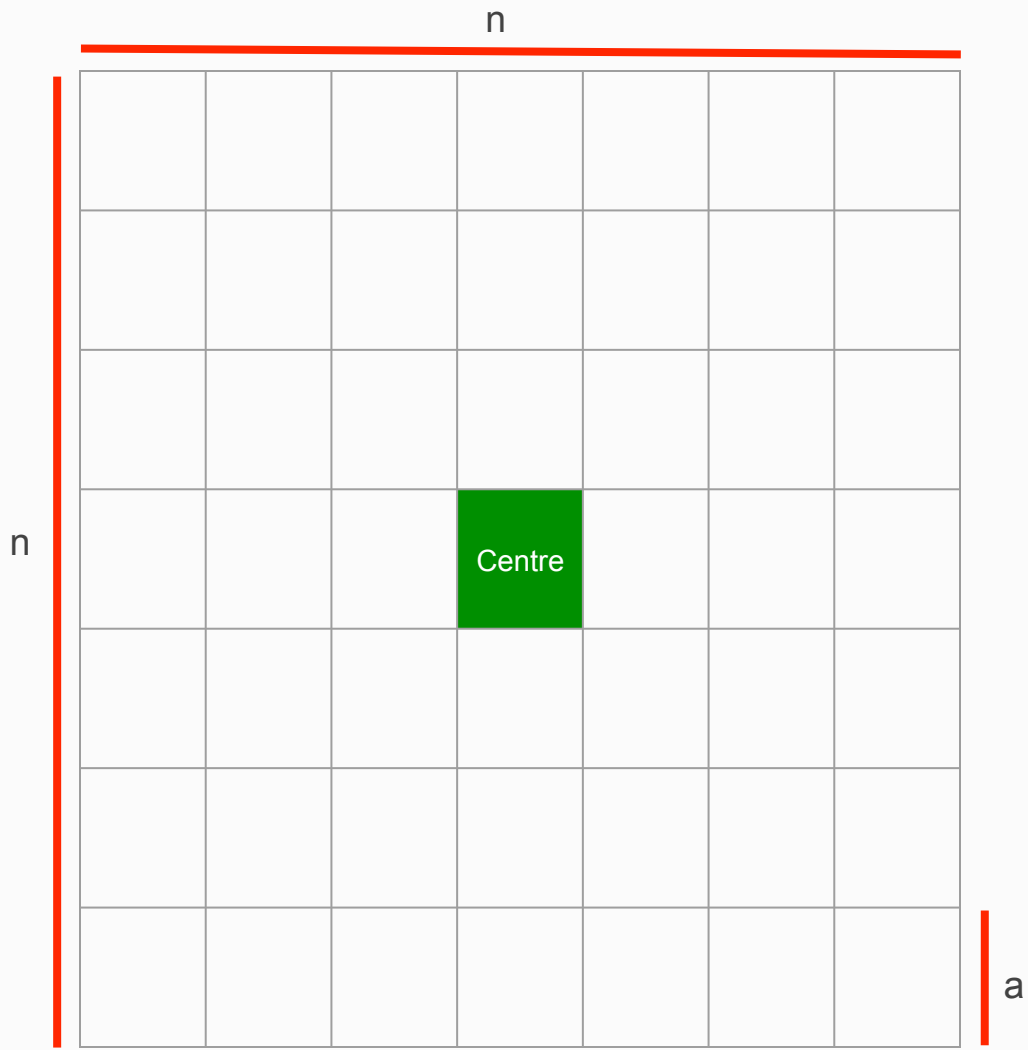
KiDOBOTiKZ™
Engineering Real Engineers

Introduction

Flood fill is an algorithm used for wall maze follower to help the robot reach the center.

In this tutorial you will be guided on the basics of wall follower and the implementation of flood fill in the robot



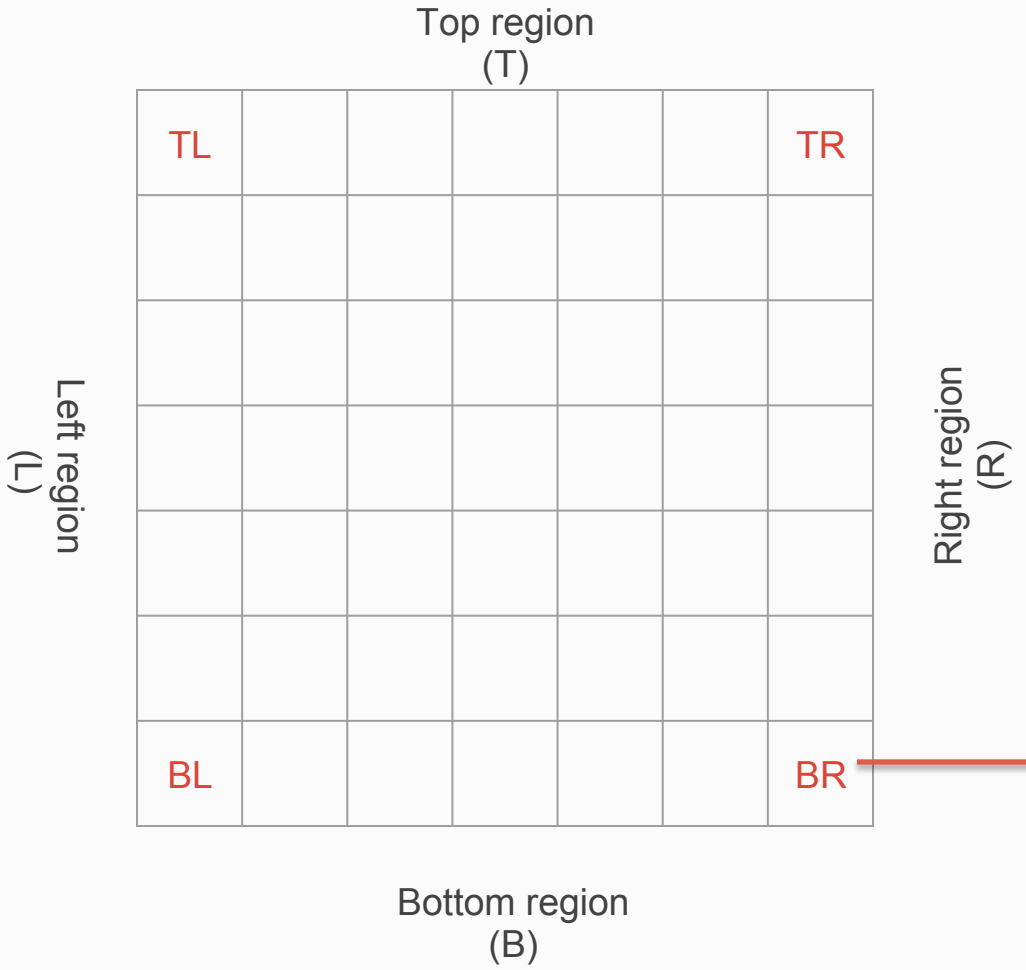


Data given before implementation

1. n x n (square)
2. a (distance of one cubic sector)

IN THIS CASE:

1. N=7
2. a usually is 15cm to 20cm (depends on the maze)



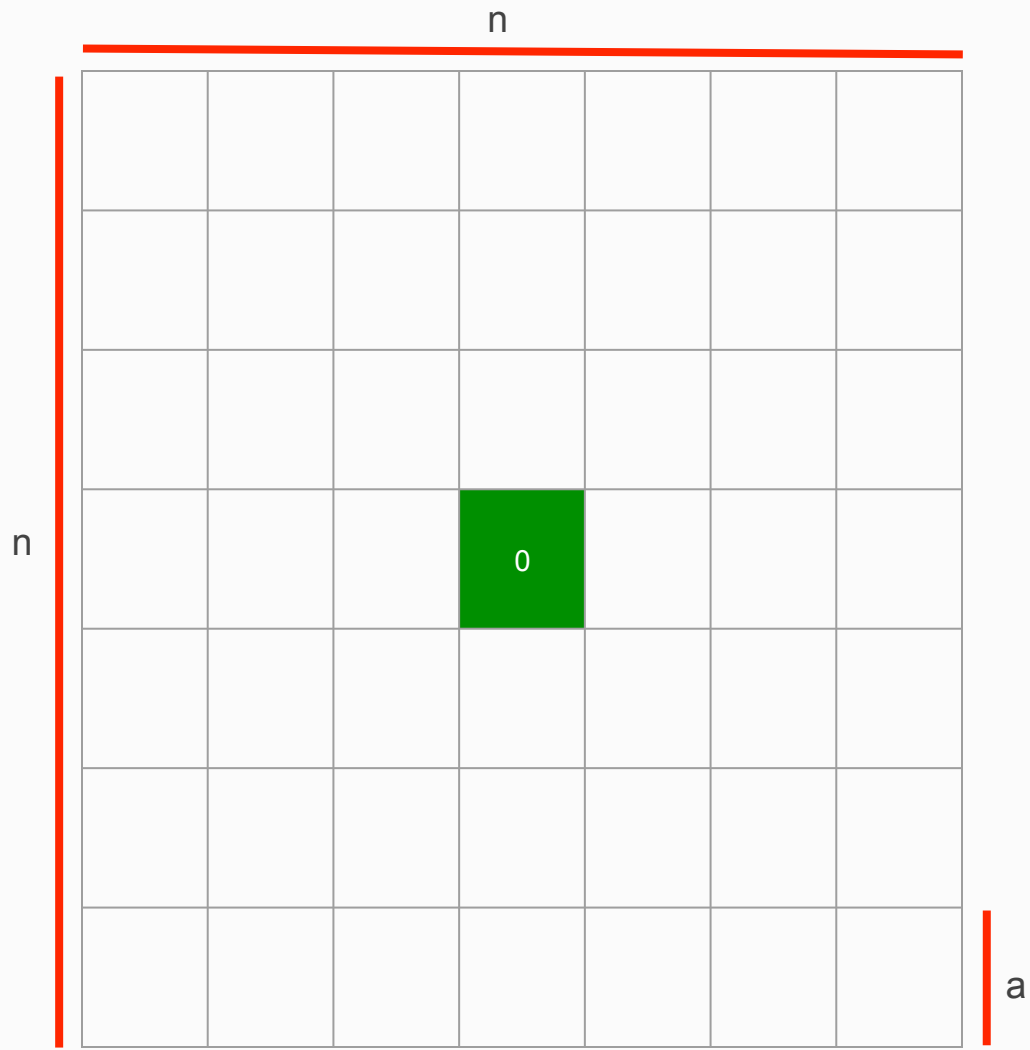
Data given before implementation

3. facing direction of the robot
4. corner at which the robot starts to move

In this case:

3. facing direction of the robot is towards top region
4. corner at which the robot starts to move is BL

Name of the corners

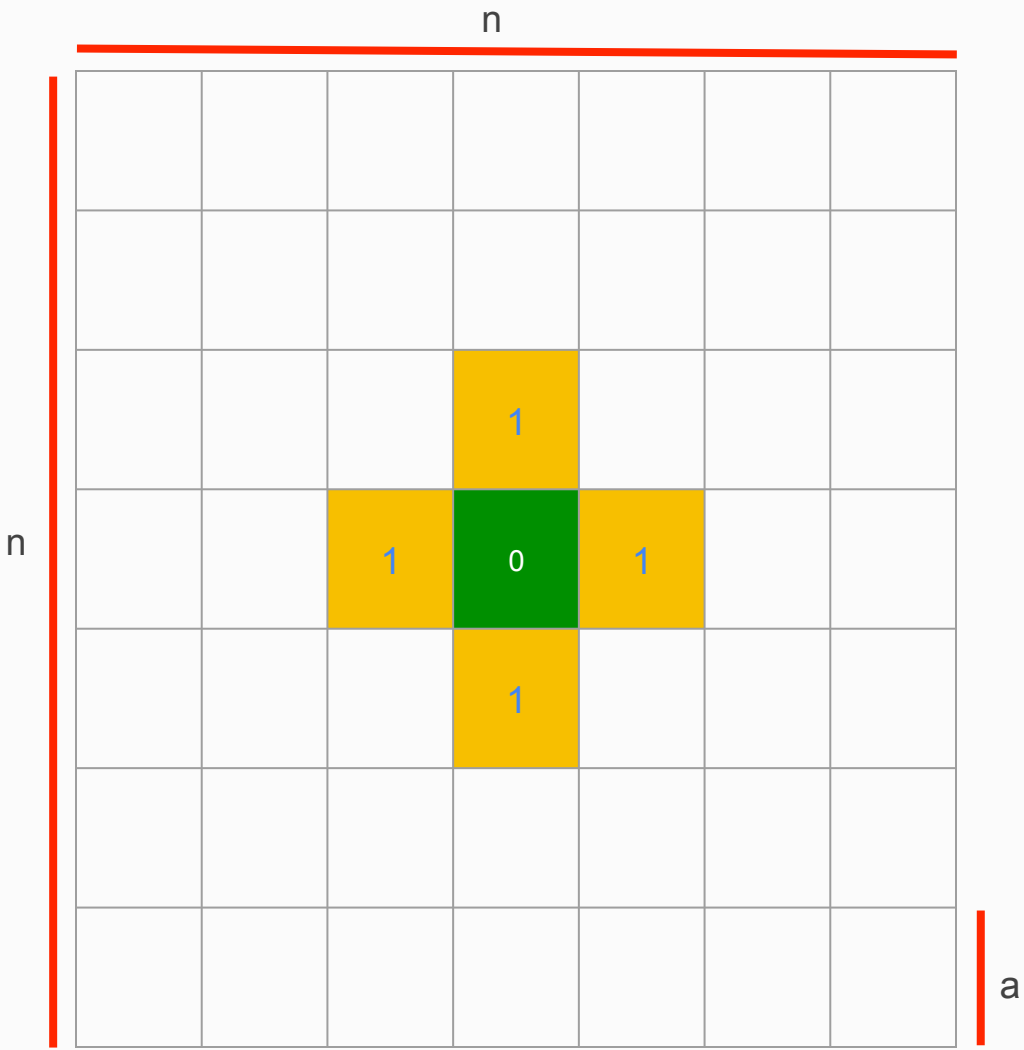


Before we start…..

We give something called weightage for the possibilities of the robot to reach the center.

STEP 1:

The centre is always given a weghtage of zero (0) – The robots aim should be to reach 0.

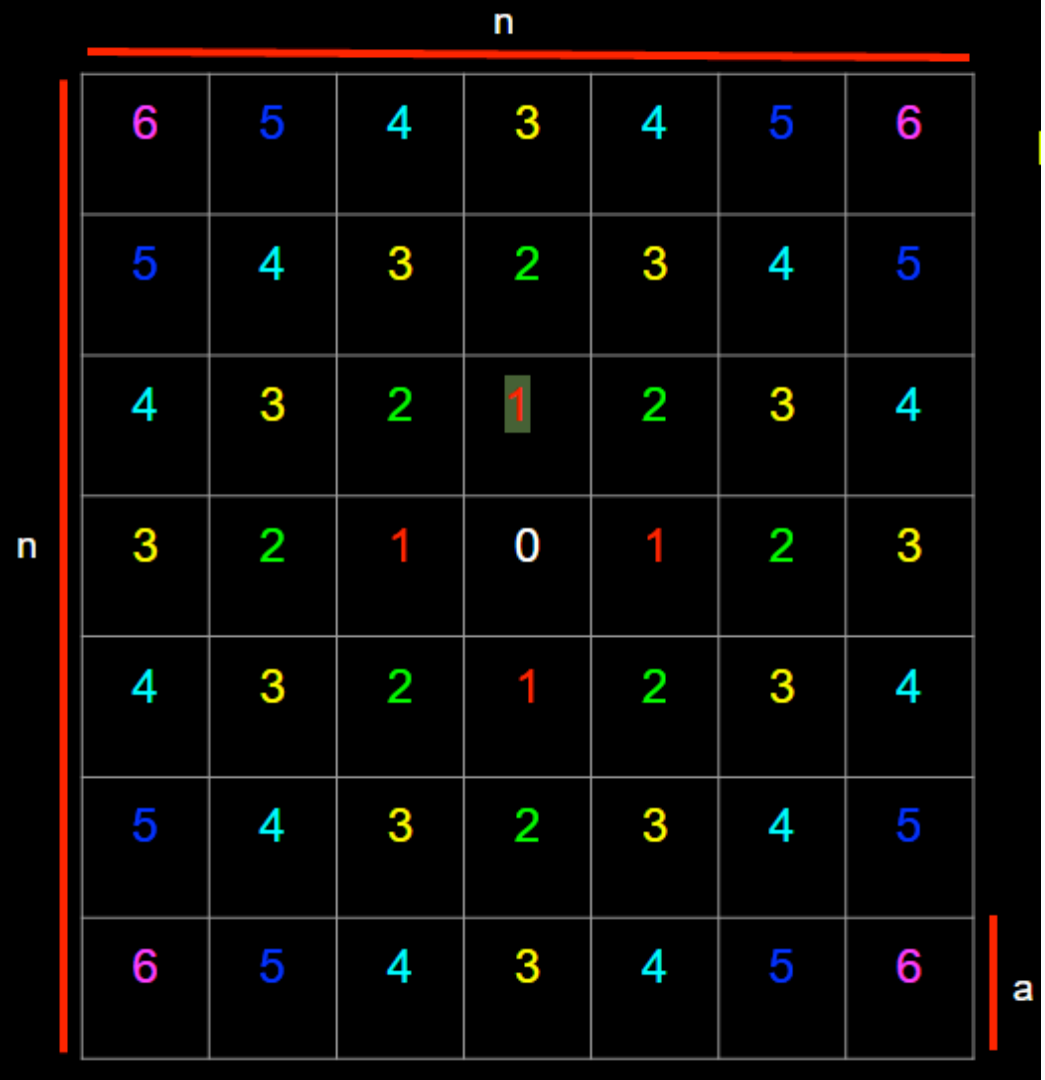


Before we start…..

STEP 2:

The cubes through which we can reach the centre is assigned the next higher value – that is, one (1)

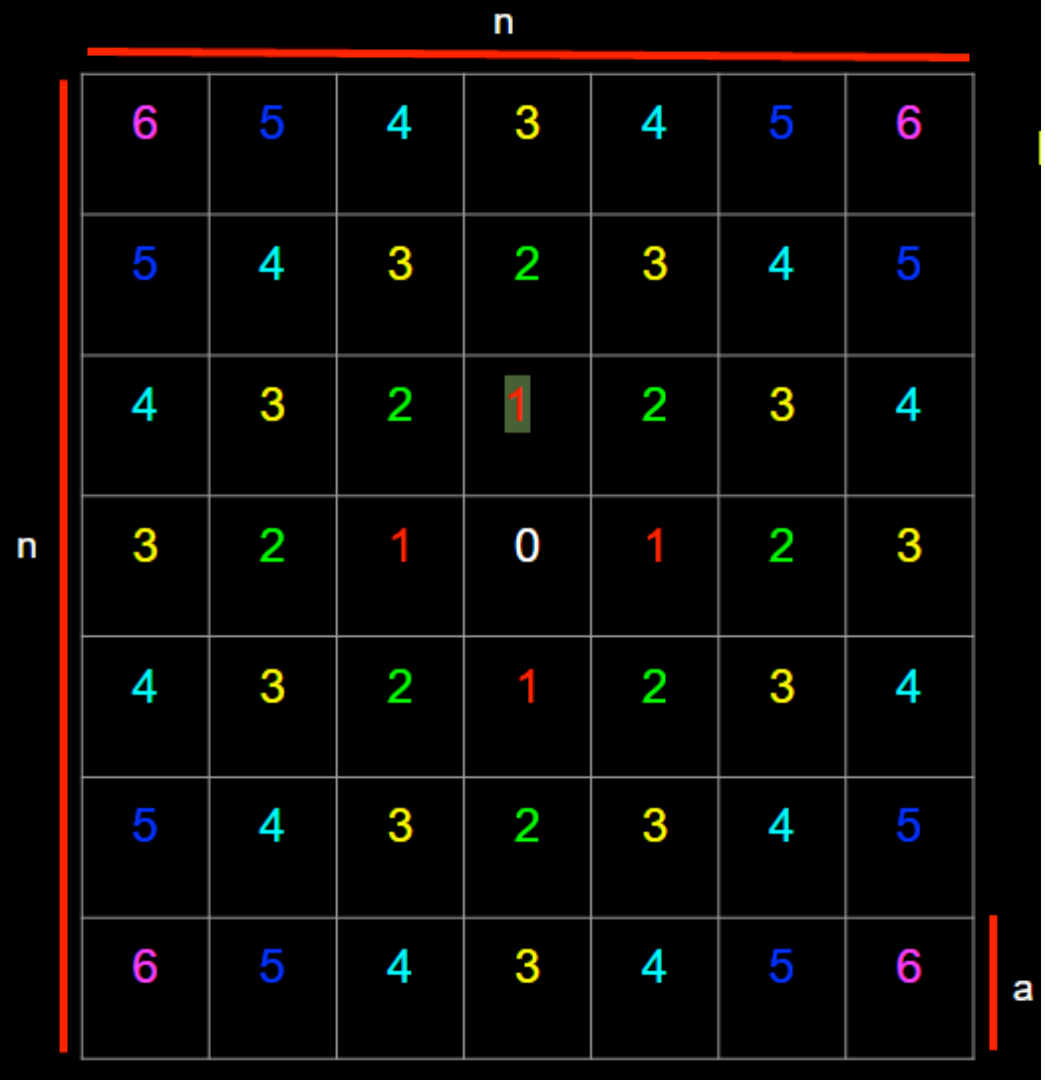
Note:
We can only enter the cubes perpendicularly and not in diagonal



Before we start…..

STEP 3:

Similarly we keep assigning the values to the cubes (increasing the value by steps of 1)

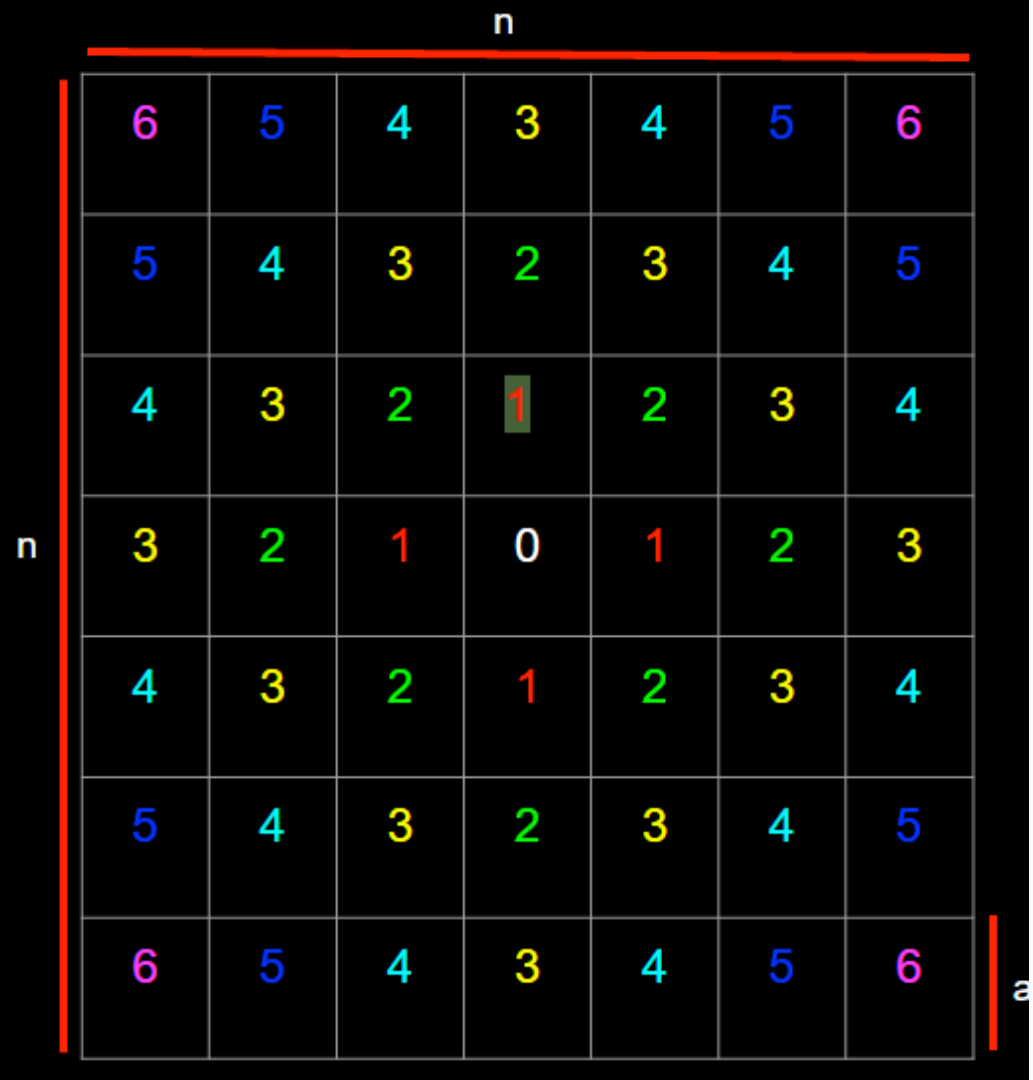


Analysis……

From the data given we can say that the starting position of our robot is always (n-1) – in this case it is (7-1 = 6).

And,

The aim of the robot is going to the centre so the main priority of the robot is to move towards the lesser number.



STOP……

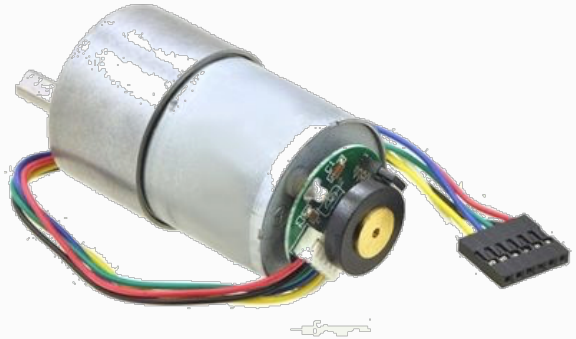
first of all how do we know whether our robot has covered ‘a’ or not (whether the robot has crossed the cube).

Thats why we use something called encoders or stepper motor.

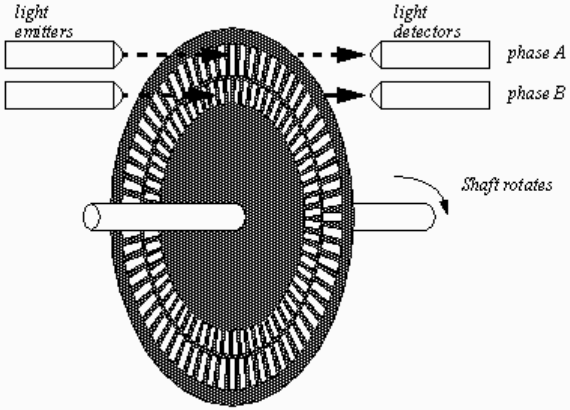
But if we use a stepper motor we need a special motor driver which increases the complexity.

KiDOBOTiKZ™
Engineering Real Engineers

What is an encoder?



An encoder may be magnetic or optical which gives an output that provides information about the motion of the shaft, which is processed elsewhere into information such as speed,distance and position.



Concepts

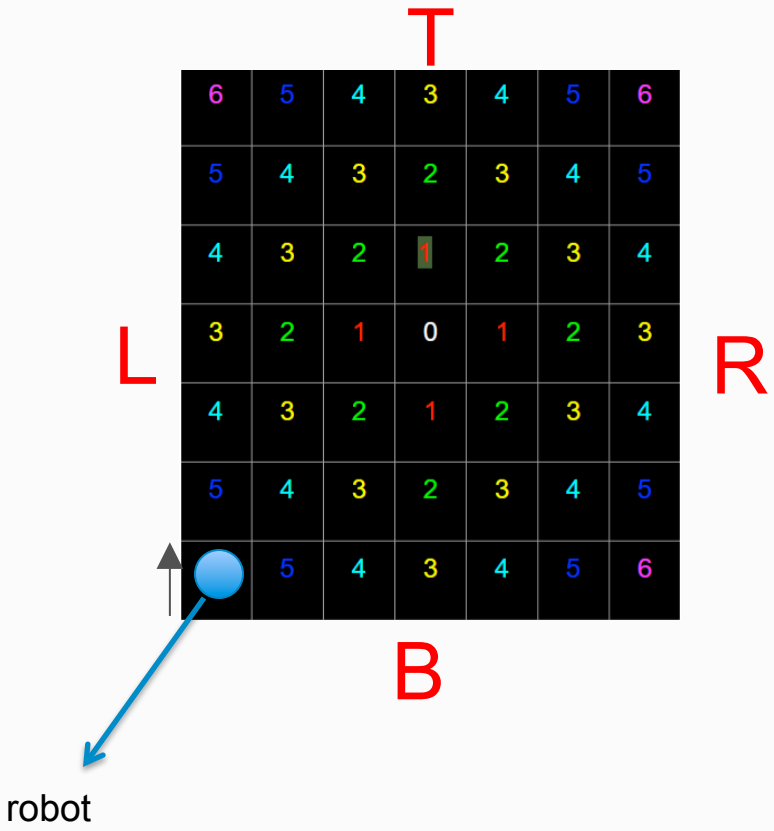
There are some concepts to be understood in your Flood fill algorithm. As your robot moves ‘a’ distance, your robot should be able to compute the foll:

1. The row and column value on which your robot is standing
2. The weightage of that position

To determine the above two, you should know the facing direction of your robot and the weightage, row value and column value of its current position

CONCEPT 1

Change of Weightage with Facing Direction

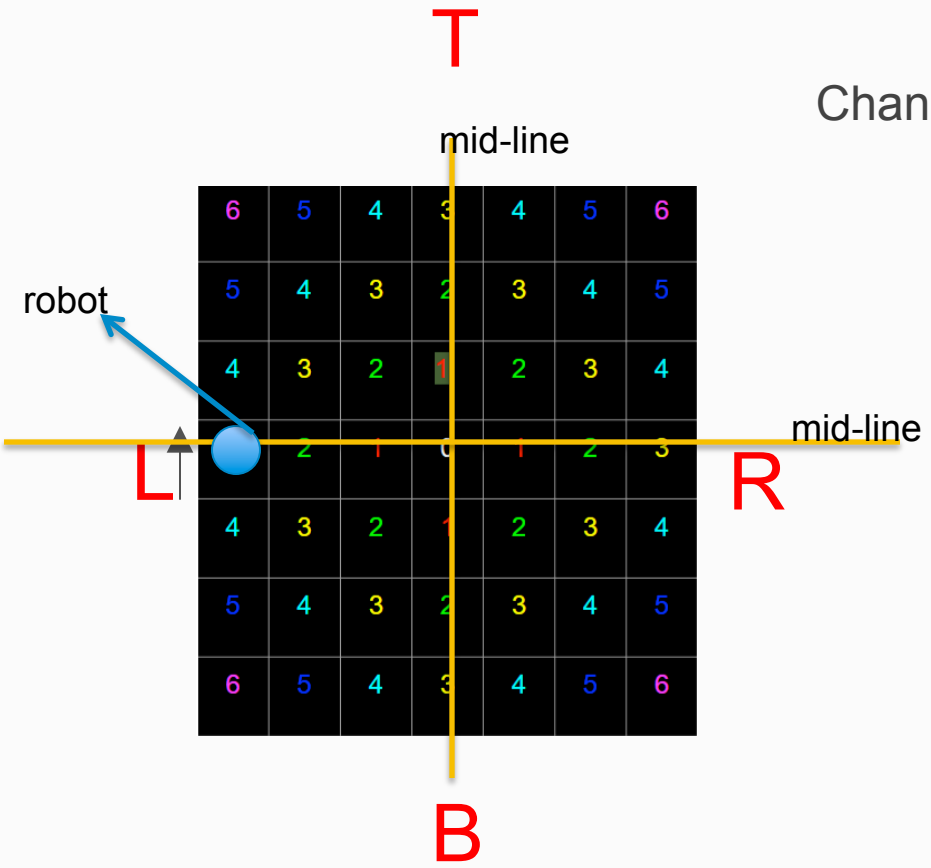


CASE 1:

Our robot is facing in the ‘T’ Direction and is at the position ‘6’, then when it moves forward we have to **reduce** the weightage of that cell

So the value will become 6-1 = 5

Change of Weightage with Facing Direction



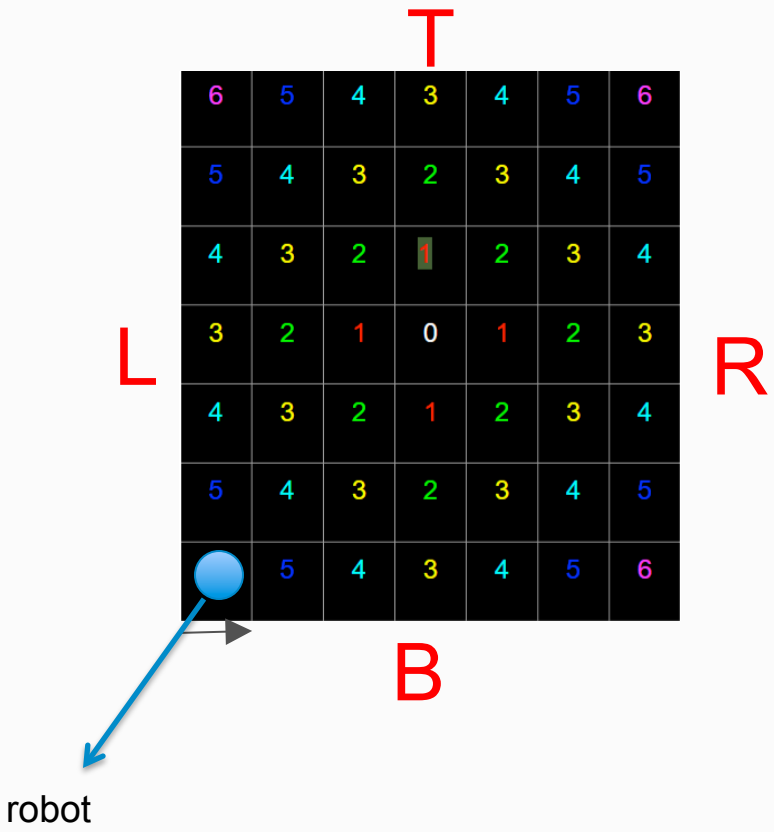
CASE 2:

Our robot is facing in the ‘T’ Direction and is at the position ‘3’, then when it moves forward we have to **increase** the weightage of that cell

So the value will become 3+1= 4

Hence, the weightage should be increased after the robot crosses the mid-line

Change of Weightage with Facing Direction

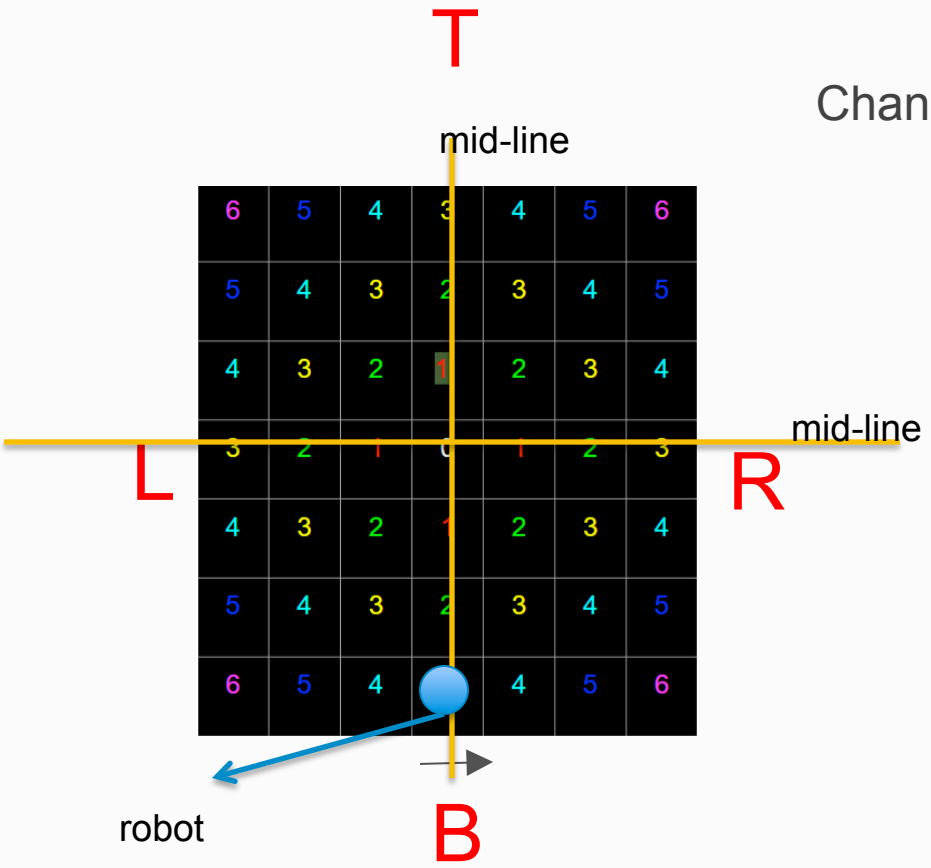


CASE 3:

SIMILARLY FOR ‘R’

Our robot is facing in the ‘R’ Direction and is at the position ‘6’, then when it moves forward we have to **reduce** the weightage of that cell

So the value will become 6-1 = 5



Change of Weightage with Facing Direction

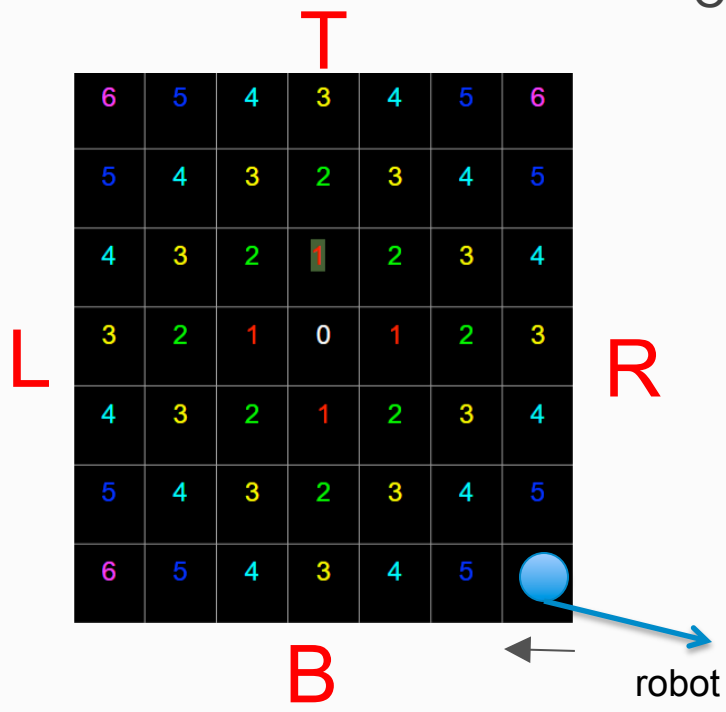
CASE 4:

Our robot is facing in the ‘R’ Direction and is at the position ‘3’, then when it moves forward we have to **increase** the weightage of that cell

So the value will become 3+1= 4

Hence, the weightage should be increased after the robot crosses the mid-line

Change of Weightage with Facing Direction

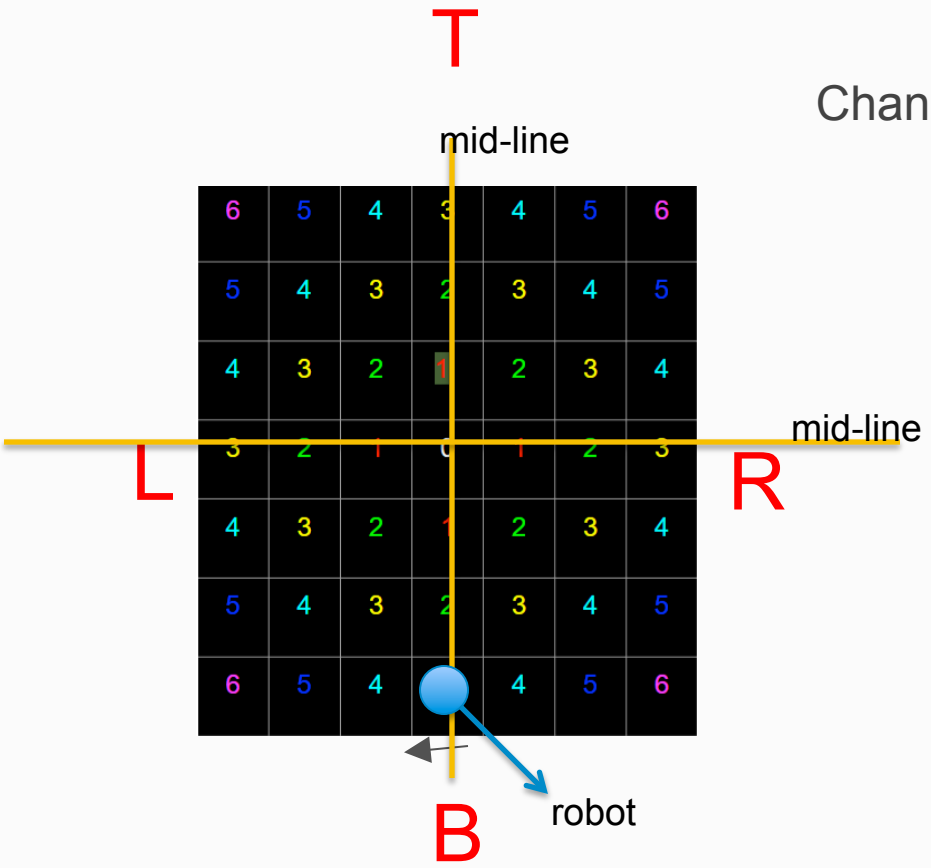


CASE 5:

Our robot is facing in the ‘L’ Direction and is at the position ‘6’, then when it moves forward we have to **reduce** the weightage of that cell

So the value will become 6-1 = 5

Change of Weightage with Facing Direction

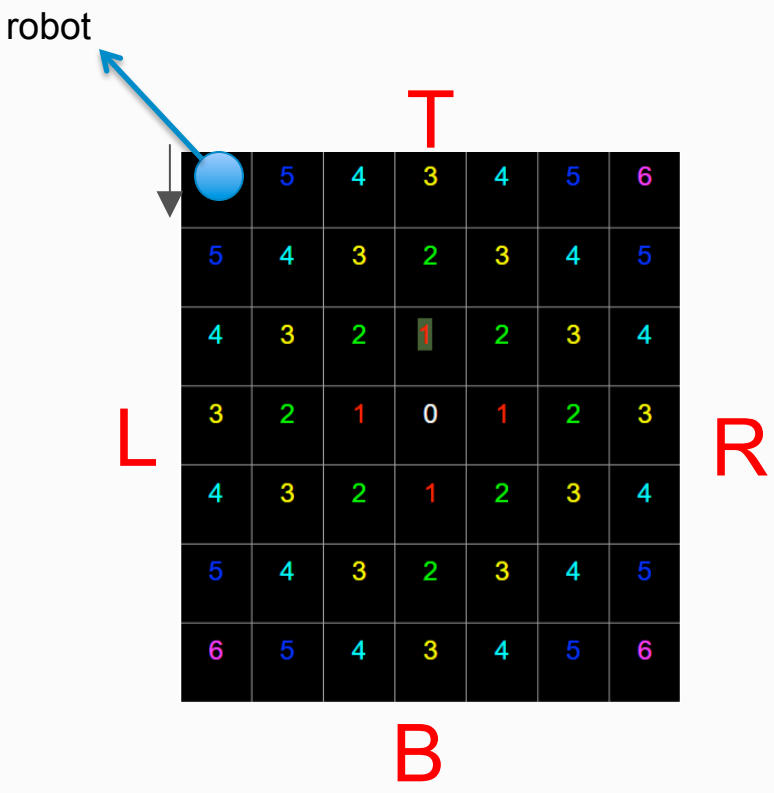


CASE 6:

Our robot is facing in the ‘L’ Direction and is at the position ‘3’, then when it moves forward we have to **increase** the weightage of that cell

So the value will become 3+1= 4

Hence, the weightage should be increased after the robot crosses the mid-line

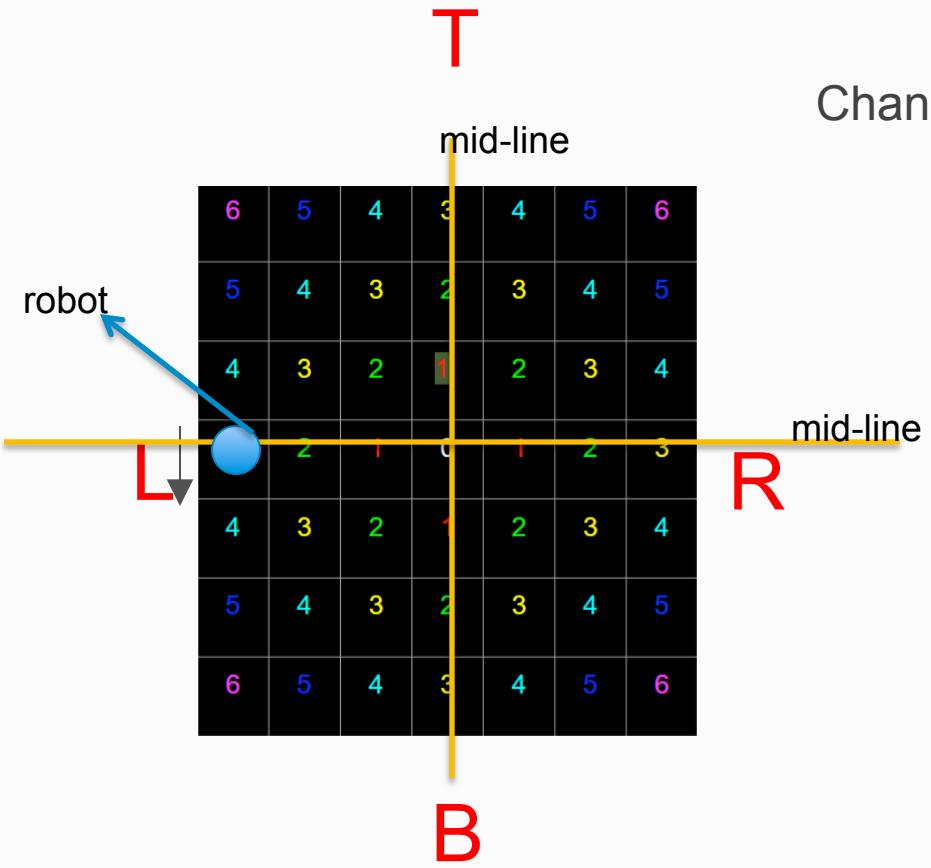


Change of Weightage with Facing Direction

CASE 7:

Our robot is facing in the ‘B’ Direction and is at the position ‘6’, then when it moves forward we have to **reduce** the weightage of that cell

So the value will become 6-1 = 5



Change of Weightage with Facing Direction

CASE 8:

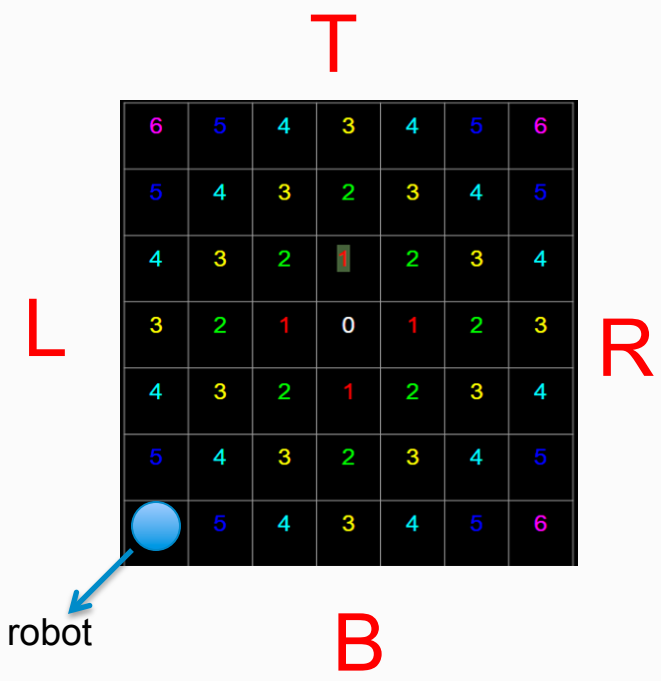
Our robot is facing in the ‘B’ Direction and is at the position ‘3’, then when it moves forward we have to **increase** the weightage of that cell

So the value will become 3+1= 4

Hence, the weightage should be increased after the robot crosses the mid-line

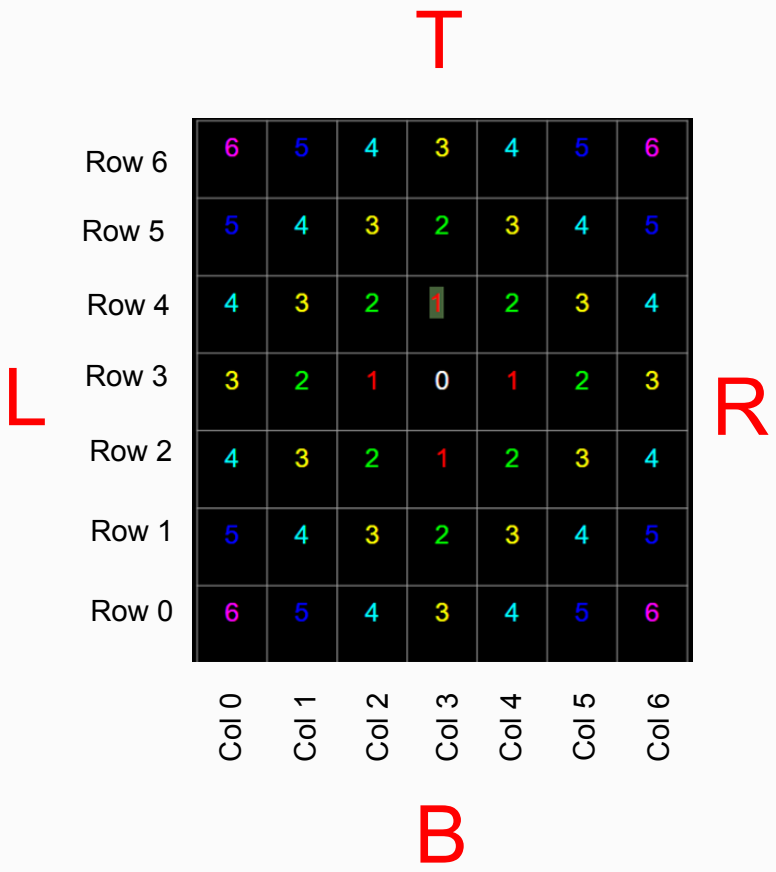
KiDOBOTiKZ™
Engineering Real Engineers

Summarizing….
Weightage value based on Facing Direction



Facing Direction	Less than (n-1)/2 (In this case, Less than 3)	Greater than (n-1)/2 (In this case, Greater than 3)
T	-	+
B	+	-
L	+	-
R	-	+

CONCEPT 2



Change of Position based on Facing Direction

Row value increases when the robot moves in the ‘T’ direction but the column value remains the same

Row value decreases when the robot moves in the ‘B’ direction but the column value remains the same

Column value increases when the robot moves in the ‘R’ direction but the row value remains the same

Column value decreases when the robot moves in the ‘L’ direction but the row value remains the same

KiDOBOTiKZ™
Engineering Real Engineers

Summarizing….
Row and Column value based on Facing Direction

FACING DIRECTION	ROW VALUE	COLUMN VALUE
T	+1	NO EFFECT
B	-1	NO EFFECT
L	NO EFFECT	-1
R	NO EFFECT	+1

KiDOBOTiKZ™
Engineering Real Engineers

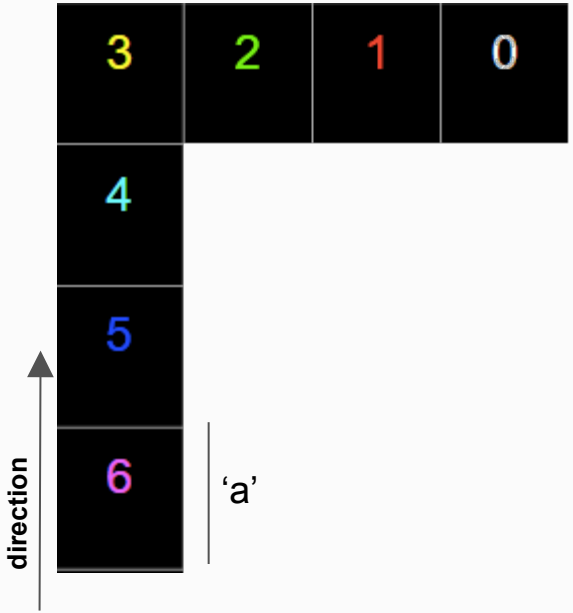
Complete Summary
– When a robot moves in its facing direction

FACING DIRECTION	ROW VALUE	WEIGHTAGE OF ITS PLACE		COLUMN VALUE	WEIGHTAGE OF ITS PLACE	
T	+1	If Row value<=(n-1)/2	-1	NO EFFECT	-NA-	
		If Row value>(n-1)/2	+1			
B	-1	If Row value<=(n-1)/2	+1	NO EFFECT	-NA-	
		If Row value>(n-1)/2	-1			
L	NO EFFECT	-NA-		-1	If Col value<=(n-1)/2	+1
					If Col value>(n-1)/2	-1
R	NO EFFECT	-NA-		+1	If Col value<=(n-1)/2	-1
					If Col value>(n-1)/2	+1

KIDOBOTIKZ™
Engineering Real Engineers

Scenario 1: (as shown in the figure)

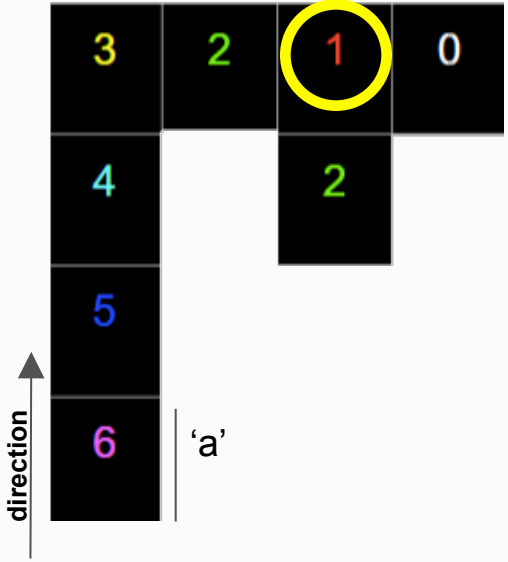
Robot Steps	Facing direction (FD)	Row Value (RV)	Col Value (CV)	Weightage (WT)
Robot starting position	T	0	0	6
Robot covers ‘a’ distance in its FD (T)	Remains the same as the robot has not turned =T	Increases by one (since FD is T) =1	Remains the same (since FD is T) =0	Decreases by 1 (since FD is T and RV<=3) =5
Same Process Continued for 2 more steps	Remains the same as the robot has not turned =T	Increases by two (since FD is T) =3	Remains the same (since FD is T) =0	Decreases by 2 (since FD is T and RV<=3) =3
Robot takes a right turn as it has only that option and then moves forward	Robot turns right and hence FD becomes =R	Remains the same (since FD is R) =3	Increases by one (since FD is T) =1	Decreases by 1 (since FD is R and CV<=3) =2
Same Process Continued for 2 more steps and centre reached	Remains the same as the robot has not turned =R	Remains the same (since FD is R) =3	Increases by two (since FD is R) =3	Decreases by 2 (since FD is R and CV<=3) =0



KiDOBOTiKZ™
Engineering Real Engineers

Scenario 2: (as shown in the figure)

Robot Steps	Facing direction (FD)	Row Value (RV)	Col Value (CV)	Weightage (WT)
Same as Scenario 1 till robot takes 5 steps (circled in Yellow)	R	3	2	1
Robot has a choice of either going forward or taking a right. Robot checks the weightage of both the cells, finds that forward cell has a lesser weightage and hence chooses to go forward	Remains the same as the robot has not turned =R	Remains the same (since FD is R) =3	Increases by one (since FD is R) =3	Decreases by 1 (since FD is R and CV<=3) =0



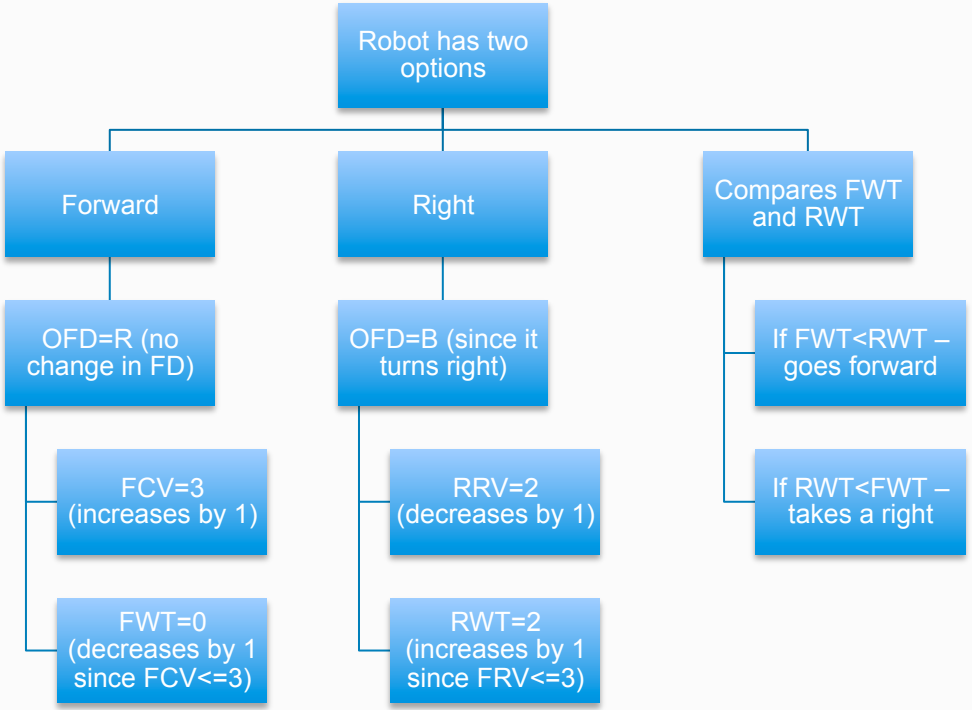
How does the robot check the weightage of the optional cells?

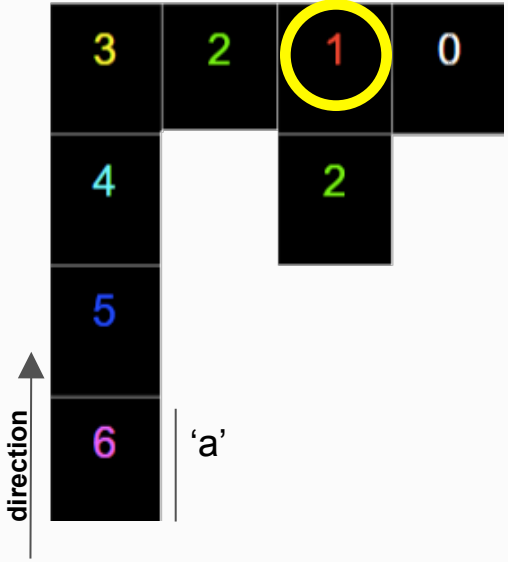
Explained in the next slide…..

KiDOBOTiKZ™
Engineering Real Engineers

Scenario 2: (as shown in the figure)

How does the robot check the weightage of the optional cells?

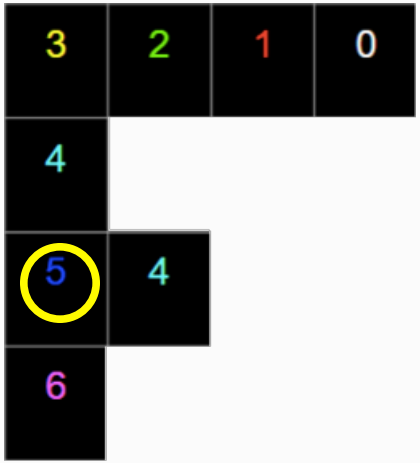




Index:
OFD – Optional Facing Direction
FCV – Forward cell's col value
FWT – Forward cell's weightage
RRV – Right cell's row value
RWT – Right cell's weightage

Scenario 3: (as shown in the figure)

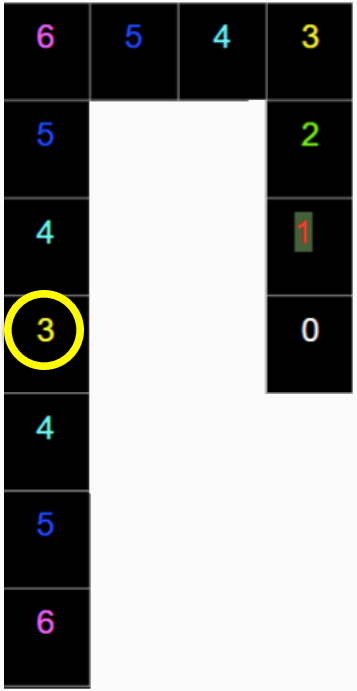
Robot Steps	Facing direction (FD)	Row Value (RV)	Col Value (CV)	Weightage (WT)
Same as Scenario 1 till robot takes 1 step forward (circled in Yellow)	T	1	0	5
Robot has a choice of either going forward or taking a right. Robot checks the weightage of both the cells, finds that both the cells have the same weightage (4) and hence chooses by its priority (Left priority robot will go straight; Right priority robot will take a right). Assuming Right priority	Robot takes a right turn, hence =R	Remains the same (since FD is R) =1	Increases by one (since FD is R) =1	Decreases by 1 (since FD is R and CV<=3) =4
Robot faces a dead-end, Hence has to take a U-turn and then go forward by 'a' distance	Robot takes a U turn, hence FD =L	Remains the same (since FD is L) =1	Decreases by 1 (since FD is L) =0	Increases by 1 (since FD is L and CV<=3) =5
Now again robot has 2 options (Left or Right). Checks weightage and chooses to turn right	Robot takes a right, hence FD =T	Increases by 1 (since FD is T) = 2	Remains the same (since FD is T) =0	Decreases by 1 (since FD is T and RV<=3) =4
Same as Scenario 1 – It reaches the centre	R	3	3	0

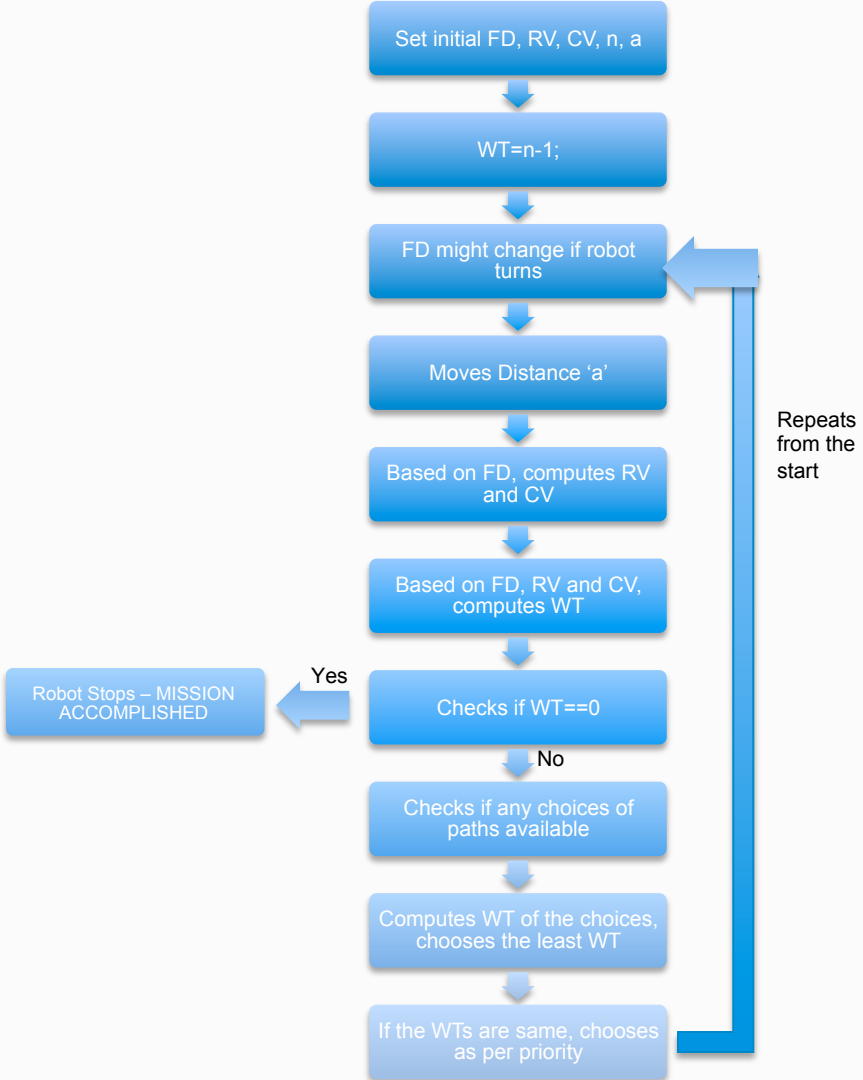


KiDOBOTiKZ™
Engineering Real Engineers

Scenario 4: (as shown in the figure)

Robot Steps	Facing direction (FD)	Row Value (RV)	Col Value (CV)	Weightage (WT)
Same as Scenario 1 till robot takes 3 step forward (circled in Yellow)	T	3	0	3
Robot has no choice but to continue forward	Remains the same, hence =T	Increases by one (since FD is T) =4	Remains the same (since FD is T) =0	Increases by 1 (since FD is T and RV>3) =4
Same Continues till the robot reaches the center	Robot would have been facing towards the bottom region =B	3	3	0





KiDOBOTiKZ™
Engineering Real Engineers

SIMPLE GENERAL ALGORITHM

Based on the previous explanations

KIDOBOTIKZ™
Engineering Real Engineers

WWW.KIDOBOTIKZ.COM