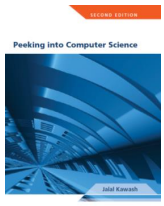
Introduction



Peeking into Computer Science

- Mandatory: Chapter 1
- Optional: None

Reading Assignment

Spreadsheet Design

Long-lasting spreadsheets

3

At the end of this section, the student will be able to:

1. Judge the design of a spreadsheet
2. Divide a spreadsheet into five sections
3. Understand the role of each section

Objectives

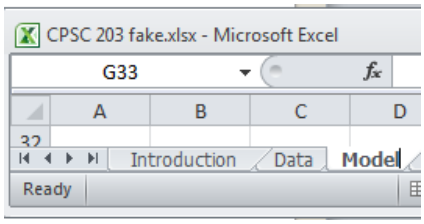
- Good character
- Easy to build
- Easy to read
- Easy to use
- Easy to change
- Error free

Spreadsheet Properties

	A	B	C	D	E	F
1		Income	Rent	Income Tax	Other Tax	Net Income
2	Dad	$ 85,000.00	$12,000.00	$ 25,500.00	$ 400.00	$ 47,100.00
3	Mom	$ 90,000.00	$12,000.00	$ 27,000.00	$ 400.00	$ 50,600.00
4	Daugther	$ 12,000.00	$ 1,000.00	$ 3,600.00	$ 400.00	$ 7,000.00
5	Son	$ 3,000.00	$ 200.00	$ 900.00	$ 400.00	$ 1,500.00
6	Totals	$ 190,000.00	$25,200.00	$ 57,000.00	$ 1,600.00	$106,200.00

Example of Poor Spreadsheet

1. Introduction:
2. Data:
3. Model:
4. Data Dictionary
5. Conclusions



Each section corresponds to an Excel tab

Spreadsheet Sections

- What are the research questions?
- How will the questions be answered?
- Summary of the findings.
- References (use only reputable ones)

Introduction Section

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Which US state should I live in?**												
2	**CPSC 203 lecture example**												
3													
4													*Januray 12, 2010*
5	**Overview**												
6	This spreadsheet was created to aid in deciding which US state we will be moving to. There are three main factors that shall affect our decision.												
7	The US state must:												
8	1. Have a warm climate.												
9	2. Have a low crim rate.												
10	3. Be close to Ottawa.												
11													
12													
13	**Data collection**												
14	The data for the aforementioned three criteria was collected as follows:												
15													
16	The average high temperatures for each state are obtained from netstate.com[1]. The fahrenheit values are then converted to Celsius using Google calculator[2].												
17													
18	Homicide rates were collected from Wainer's "Graphic Discovery"[3].												
19													
20	The distance to Ottawa was found by using Google Maps[4] to get the distance from the state capital to Ottawa.												
21													

Example Intro

- Use APA or other format for citation:
  - http://pages.cpsc.ucalgary.ca/~tamj/2007/203W/assignments/references.html (if you have no idea for a format)
  - J. Nevison, *The Elements of Spreadsheet Style,* Prentice-Hall, 1987 (for these notes)
  - Sometimes the ideas are so common or so simple that a citation is unneeded.
  - Other times listing the sources can be quite useful.
    - Basis of assumptions, places to look for more information.

JT's Extra, Introduction: References

	A	B	C	D	E	F	G	H	I	J	K	L
21												
22												
23												
24	**Table of Contents**											
25	Model:	Explains the calculations conducted in the Data tab, as well as listing all formulas used.										
26	Data Dictionary:	Explains each set of data so that its type and description are given.										
27	Data:	Contains the raw data and calculations.										
28	Dashboard:	Contains charts based on the Data tab.										
29												
30												
31	**References**											
32	1. Nstate LLC. (2009, September 24). Retrieved September 25, 2009, from netstate.com: www.netstate.com											
33	2. Google. (2009, September 25). Google Calculator. Retrieved September 25, 2009, from google.com: www.google.com											
34	3. Wainer, H. (2004). Graphic Discovery: A Trout in the Milk and Other Visual Adventures. Princeton: Princeton University Press.											
35	4. Google. (2009, September 25). Google Maps. Retrieved September 25, 2009, from http://maps.google.com											
36												

Example Intro

- Raw data
  - Example (from "Problems and spreadsheets" 203 notes, finding a place to live in the U.S.): temperature, homicide rates, distance.
  - Use trusted sources of data (e.g., Stats' Can over some random blog)
- Calculations
  - Custom formulas you created
    - e.g. `= D3 + (E1 – F1)`
  - Built-in ones e.g., `SUM()` or even more complex like '`if`'

JT's Extra:
It's an important section but the whole sheet should not consist just of data and calculations

Data Section

	A	B	C	D	E
1	**State**	**Avg high temperature**	**HomicideRate**	**Time to Ottawa**	**Objective Value**
2	Alabama	33	12	20.37	49.826
3	California	33.44	8.8	41	47.432
4	Connecticut	29.33	3.9	7.12	55.425
5	Georgia	33.44	8.7	18.26	52.03
6	Illinois	30.6	9.8	15.19	51.242
7	Kansas	33.8	6.1	21.7	52.75
8	Louisiana	34	16.1	25	47.15
9	Maine	26	1.8	17.13	53.474
10	Maryland	30.6	10.9	9.13	51.904
11	Minnesota	28.55	2.8	19.7	53.225
12	Mississippi	33.6	14.2	22.27	48.526
13	Nebraska	31.94	3.9	20.11	53.61
14	New Hampshire	28.11	2.2	6.13	56.107
15	New York	29.61	6.3	5.25	54.683
16	South Dakota	30.27	3	25	52.581
17					
18					
19	**Weights**				
20	Temperature	30%			
21	Homicide	50%			
22	Time to Ottawa	20%			
23					

Example Raw Data

- Model includes assumptions and calculations
- Explain the model:
- Should provide three levels of explanation:
  - Explain the values appearing in the model
  - Explain tricky formulas
  - Lists all formulas explaining in steps with all assumptions stated clearly.

Model Section

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**Model**													
2														
3	There are three factors adding to our final preference: The average high temperature, the homicide rate and the time to ottawa.													
4	However, the three factors are not of the same importance. Thus, each of the factors is given a *weighting* based on how important we think it is.													
5														
6	The weight were decided as follows:													
7	AvgTempWeight		30%											
8	HomicideRate		50%											
9	Close to Ottawa		20%											
10														
11	Since lower homicide rates are clearly preferred, the weight is applied to 100 - HomicideRate.													
12	Also, since a smaller time to Ottawa is preferred, the Time to Ottawa needs to have a negative effect on the final objective value. Thus, a longer tr													
13														
14	The final formula for the Objective Value is thus:													
15														
16	**Formula for Objective Value**		AvgTemp*AvgTempWeight + (100 - HomicideRate) * HomicideWeight - TimeToOttawa*TimeOttawaWeight											
17														

Example Model

Weak

INFLATE A 1 January 1510 J. Horner
(C) Copyright 1983 John M. Nevison

Find the profit margin in an inflationary world where raw material costs, labor costs, and prices each grow at a different rate.

Growth Rate	1.03	1.15		1.07		
Year	Raw mat	Labor	Total cst	Price	Profit	Margin
1510	56.00	21.00	77.00	100.00	23.00	23.00%
1511	57.68	24.15	81.83	107.00	25.17	23.52%
1512	59.41	27.77	87.18	114.49	27.31	23.85%
1513	61.19	31.94	93.13	122.50	29.37	23.98%
1514	63.03	36.73	99.76	131.08	31.32	23.90%
1515	64.92	42.24	107.16	140.26	33.10	23.60%

(From "ELEMENTS OF SPREADSHEET STYLE" by JOHN M. NEVISON)

JT Extra: Poor Model

Strong

INFLATE B 1 January 1510 J. Horner
(C) Copyright 1984 by John M. Nevison

Test pricing in an inflationary world where different costs growing at different rates affect the margin (% profit). By varying the price growth rate the user can attempt to preserve a certain margin in some future year.

Contents: (each section is a named range)
INTRO Introductory material: Title, description, and contents.
INIT Initial data
MODEL Model

Initial Data:
1510 Starting Year

Cost structure
$56.00 Raw material cost
$21.00 Labor cost
$100.00 Price

Growth rates
1.03 Raw material growth rate
1.15 Labor growth rate
1.07 Price growth rate

Model

YEAR	MATERIAL	LABOR	TOTLCOST	PRICE	PROFIT	MARGIN
1510	56.00	21.00	77.00	100.00	23.00	23.00%
1511	57.68	24.15	81.83	107.00	25.17	23.52%
1512	59.41	27.77	87.18	114.49	27.31	23.85%
1513	61.19	31.94	93.13	122.50	29.37	23.98%
1514	63.03	36.73	99.76	131.08	31.32	23.90%
1515	64.92	42.24	107.16	140.26	33.10	23.60%

(From "ELEMENTS OF SPREADSHEET STYLE" by JOHN M. NEVISON)

JT Extra: Clearer Model

- When providing instructions:
  - A numbered list can sometimes be a tremendous improvement over text bunched into a single paragraph
- Evaluating clarity:
  - Write out the instructions and then see if an outsider can follow them.

JT's Extra: Instructions, am I clear enough now?

- Explains the meaning of your data (raw data or calculated data)
- For each field/piece of data there will be 4 columns in the data dictionary
  - A name: brief but clear description or label
  - A field type: categorization, raw data, column calculation, row calculation
  - A data type: text, number, integer, percentage, etc…
  - A sheet/cell reference: the sheet where the field is and where in the sheet (cell range)

Data Dictionary

	A	B	C	D	E	F	G
1						Data Dictionary	
2							
3							
4	**Name**	**Field Type**	**Data Type**	**Sheet/Cell Reference**			
5	State	Categorization	Text	Data!A2:A16			
6	Avg high temperature	Raw	Float	Data!B2:B16			
7	Homicide Rate	Raw	Float	Data!C2:C16			
8	Time to Ottawa	Raw	Float	Data!D2:D16			
9	Objective Value	Row Calculation	Float	Data!E2:E16			
10	Weights	Raw	Percentage	Data!B20:B22			
11							

Example Data Dictionary

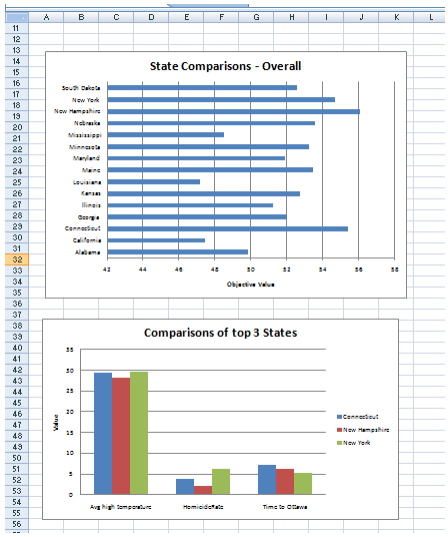
- Summarizes important data (trends found your research) into 3 charts (not pie)
- Add appropriate annotations:
  - charts have appropriate titles,
  - axes are also titled and have an appropriate scale (include a legend if necessary).
- Conclusions should be tied to the original research questions: what, how, findings of your research, reference list (citations).

Conclusions Section

- Use as little 'ink' as possible:
  - Obviously you should avoid redundancy i.e. each chart should analyze a different trend
  - But go beyond this - don't add clutter, make annotations concise as possible but still conveying a clear message
  - Also make sure that the message is represented using the most effective mechanism possible (see my extra notes on 'visualizing data')

Conclusions section (2)

- Visual Charts and conclusions

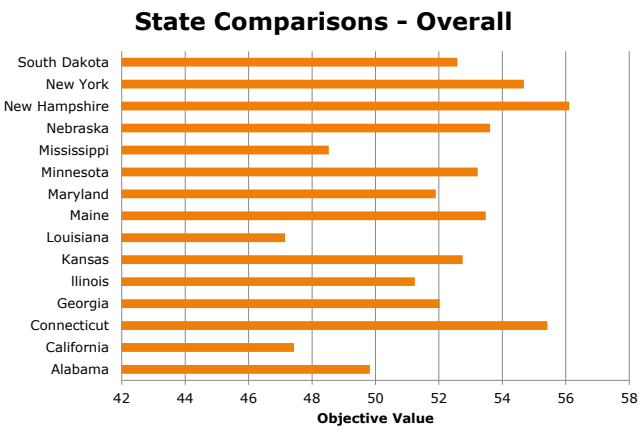


Dashboard

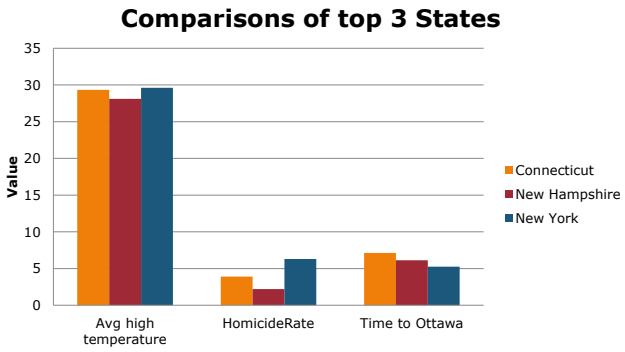
- "Essentially, a dashboard report is a way to visually present critical data in summary form so that you can make quick and effective decisions, in much the same way that a car dashboard works."[1]

1 One source (by no means the only or definitive one):
http://office.microsoft.com/en-us/excel-help/the-power-of-dashboard-reporting-with-excel-HA001226127.aspx

5) JT Extra: What Is A Dashboard



Example Dashboard



Example Dashboard

Abstraction

The power to simplify

27

- Problems (and solutions) can be overwhelming
  - Too many details
- Abstraction: hiding irrelevant details

Abstraction

- Library
- Very complex
- To an end-user, it can be abstracted by the circulation desk

Abstraction Example

- Car
- Also very complex
- To driver, it can be abstracted by how to operate it
- To a passenger, it is simply a commuting device

Abstraction Example

JT: Robot example (minor modifications to the Kawash example)

- Develop an algorithm for a simple robot (similar in movement capabilities to a Roomba™).
- Movement:
  - The robot can move forward one distance unit (a 'square').
- Rotation:
  - If forward motion is not possible then the robot can rotate left or right by 90 degrees.
- Short range sensors:
  - One is mounted forwards, the other is mounted on the right.
  - The sensors check for obstacles in the next square.

Another Problem: Robotic Movement[1]

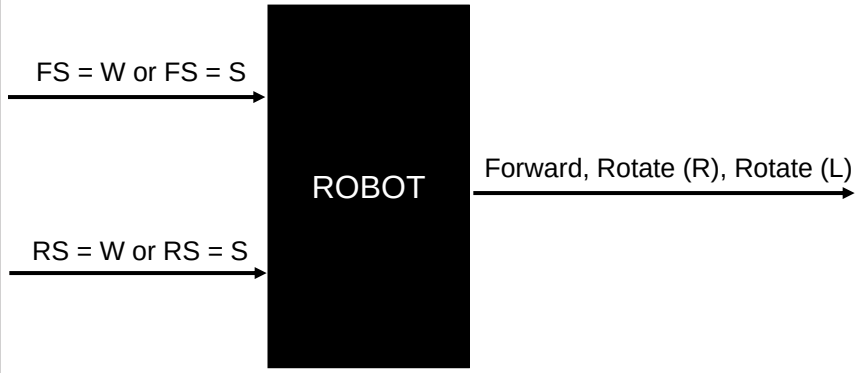
1 From "Peeking into Computer Science" and the lecture notes of Jalal Kawash

- What does the robot need to do:
  - Find a wall/obstacle.
  - Hug the wall, indefinitely moving forward.
- Input:
  - Whatever is detected by the sensors (front, right).
- Output:
  - The robot's movement

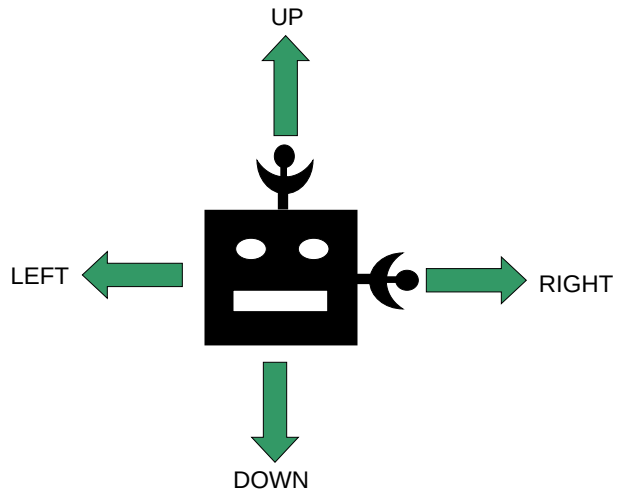
Specifying The Problem

- Since the robot can either move onto a square that's empty or avoid a square that is occupied, the world can be simplified into two cases:
  - The destination square is empty: 'space'.
  - The destination square is not empty: 'wall' (contains a wall, furniture, person, pet etc.)
    - Details about exactly why the destination isn't empty isn't important so simplify the problem.

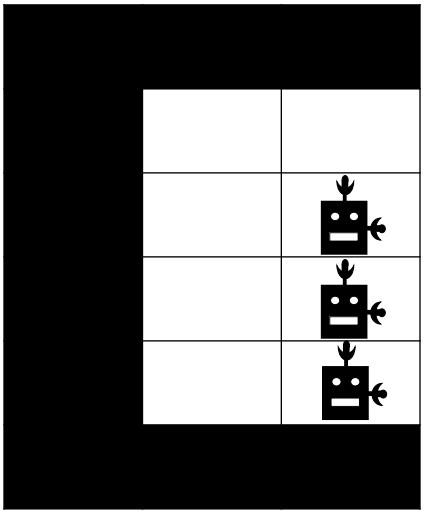
The Contents Of The Robot's World



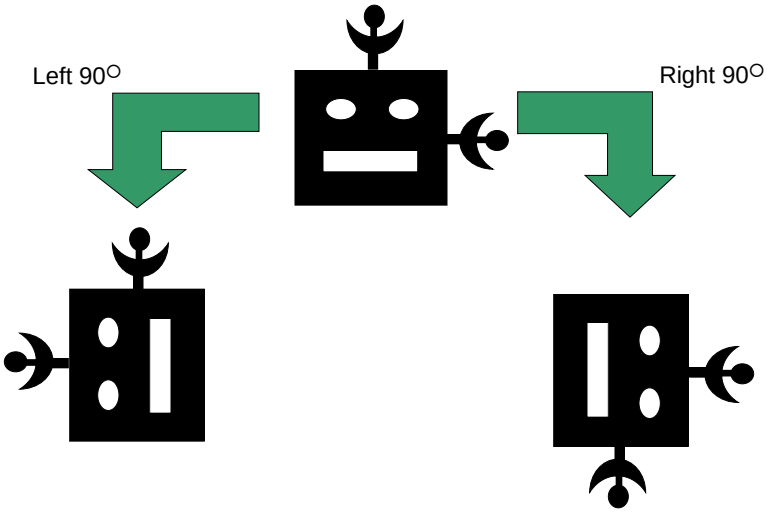
Inputs/Outputs Based On The World



Robot's Orientation



Robot: Moving Forward



Robot: Rotations

- Search for the wall
- Once found, keep it to the robot's right
- Move forward
  - Each move, make sure the wall is still to the robot's right.
  - (This means there should be a space in front and the wall to the right).
- Robot's modes:
  - Search for the wall
  - Hug the wall

Solution: The Generic Algorithm For Movement

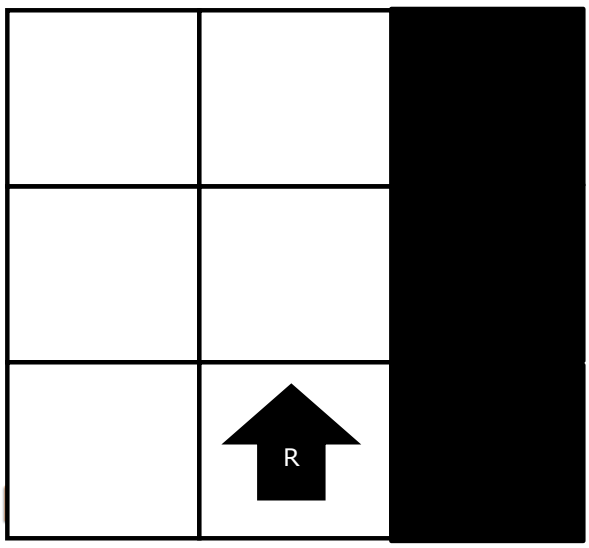
Repeat the following steps, until this phase is done (wall found, change to the wall hugging mode)

- If **RS = W**, then done this phase
  - Right sensor detects a wall
- If **FS = W**, then **L,** done this phase
  - Front sensor detects a wall, rotate left
- If **FS = S**, then **F**
  - Right sensor senses a space, take a step forward

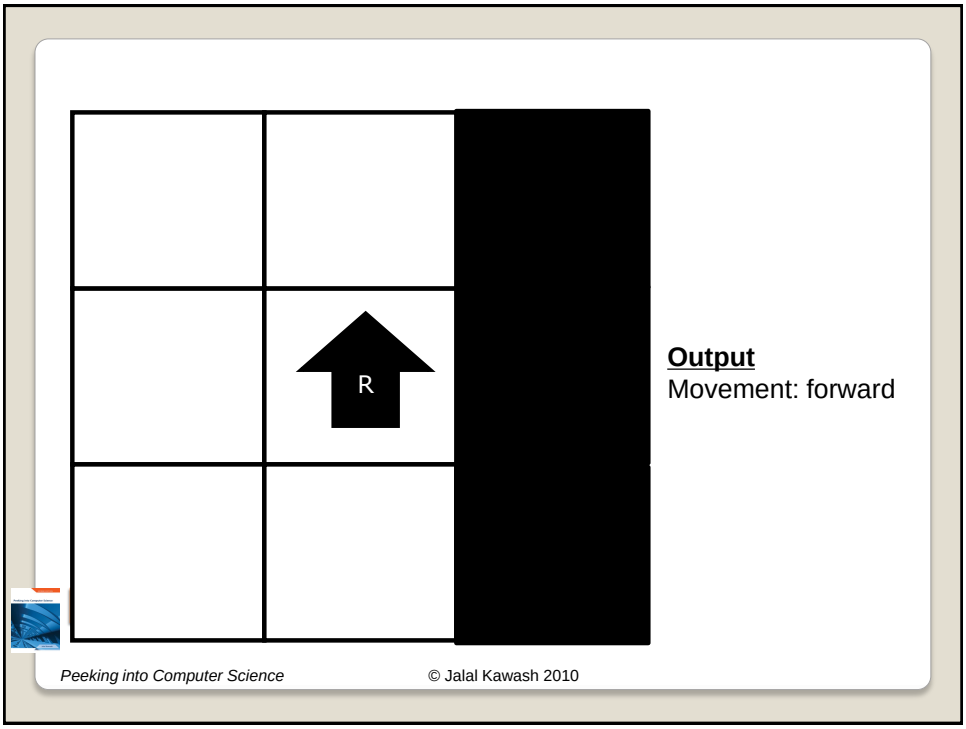
Algorithm: Search For The Wall

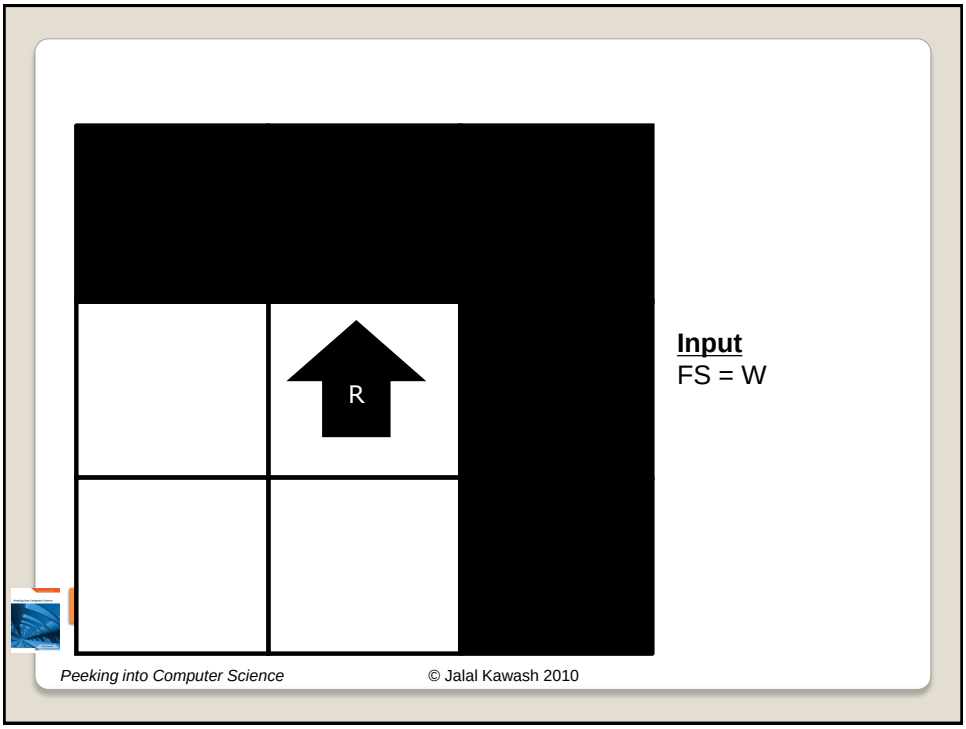
- Need to make sure that the wall is not "lost" during movement.
- Complexity: all cases must be considered.

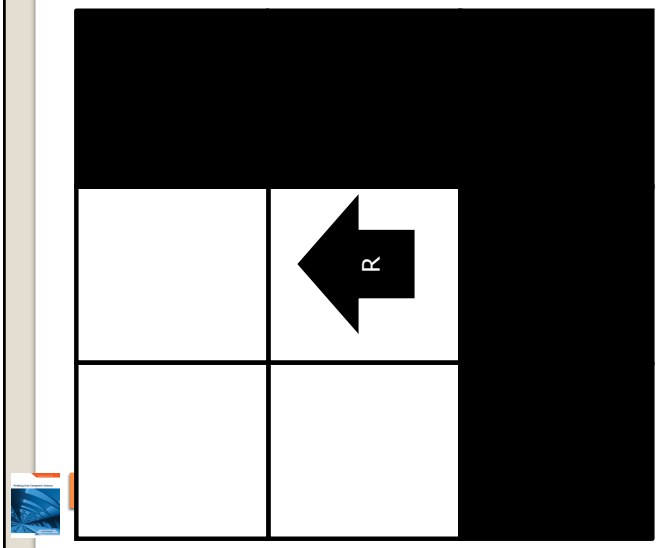
Algorithm: Hug The Wall



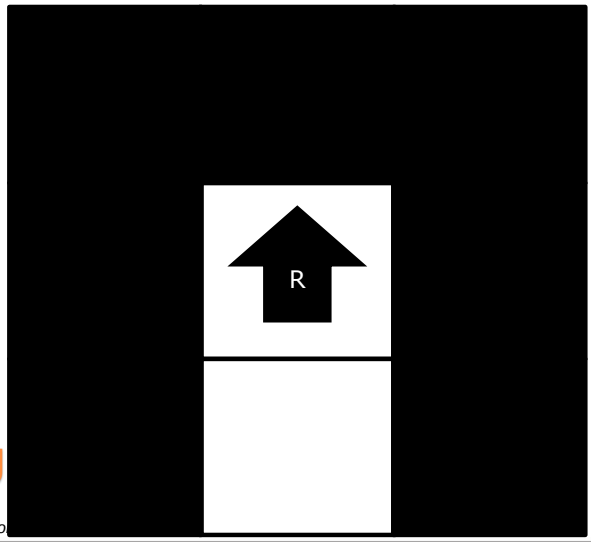
Input
RS = W
FS = S


R
Output
Movement: forward
Peeking into Computer Science
© Jalal Kawash 2010


R
Input
FS = W
Peeking into Computer Science
© Jalal Kawash 2010

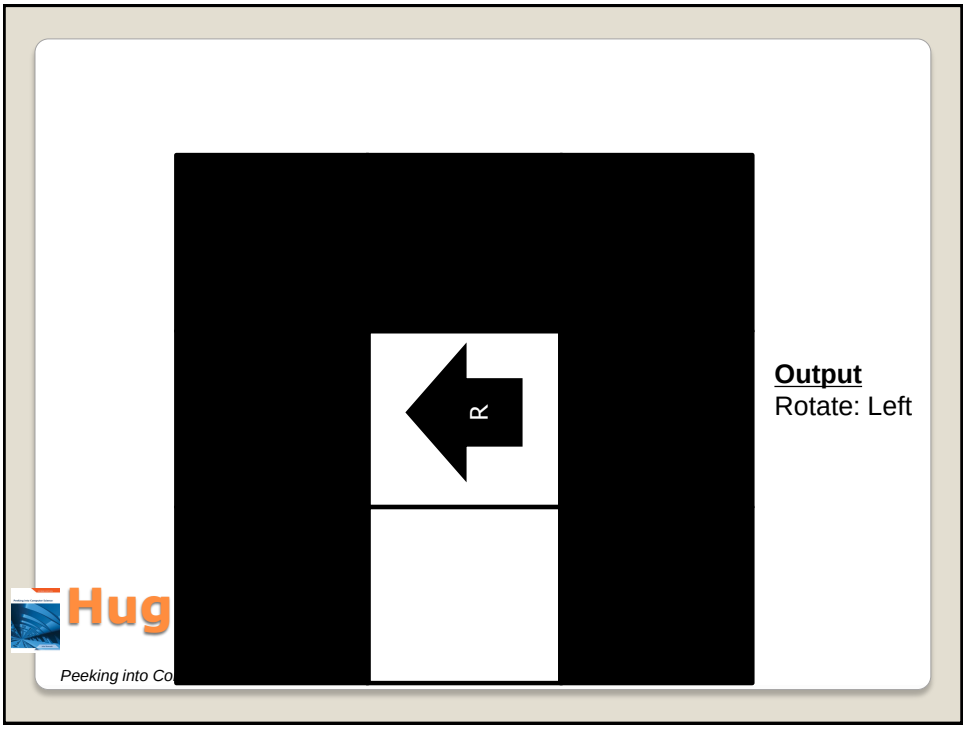


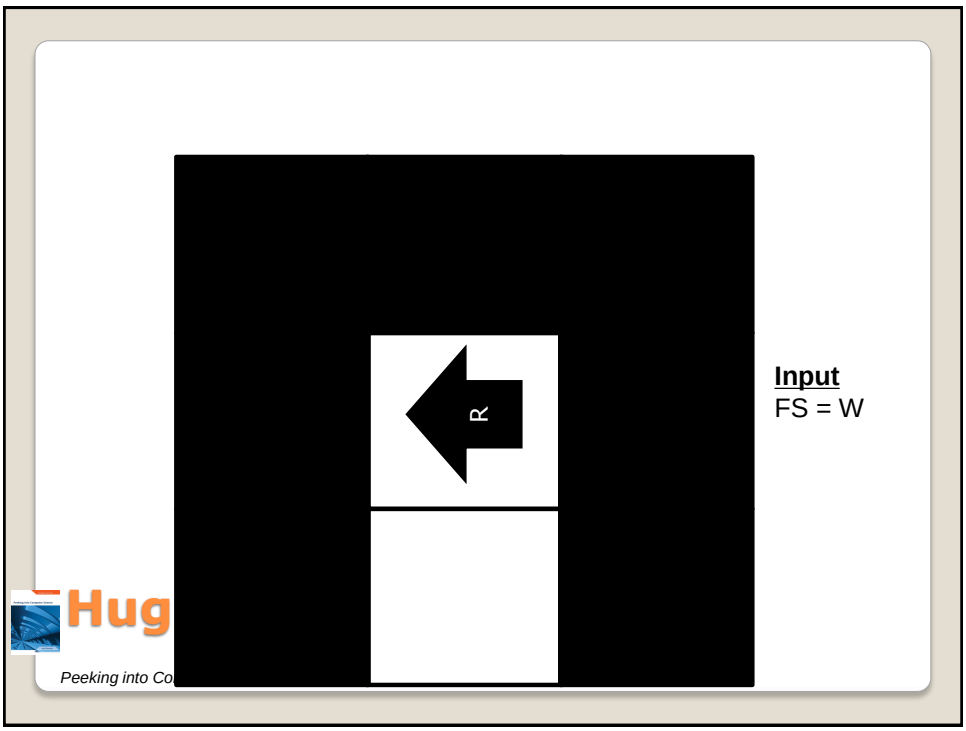
Output
Rotate: left

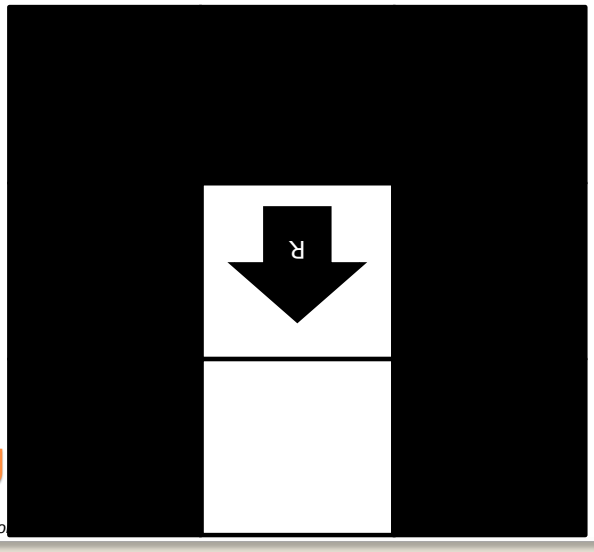


Input
FS = W

Hug

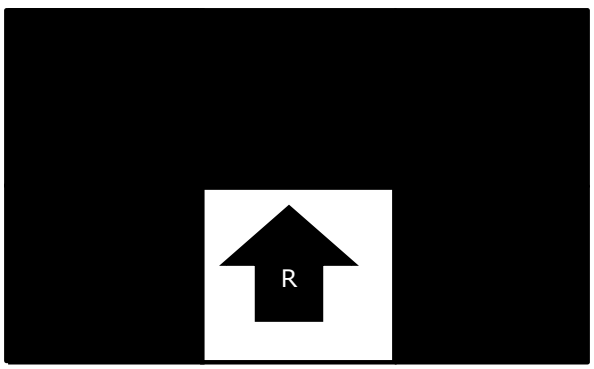

R
Output
Rotate: Left
Hug
Peeking into Co


R
Input
FS = W
Hug
Peeking into Co



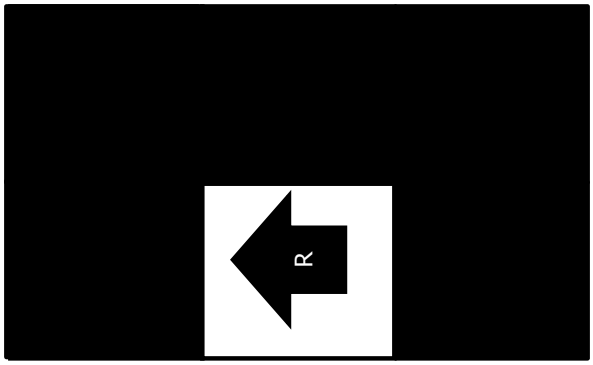
Output
Rotate: Left

Hug



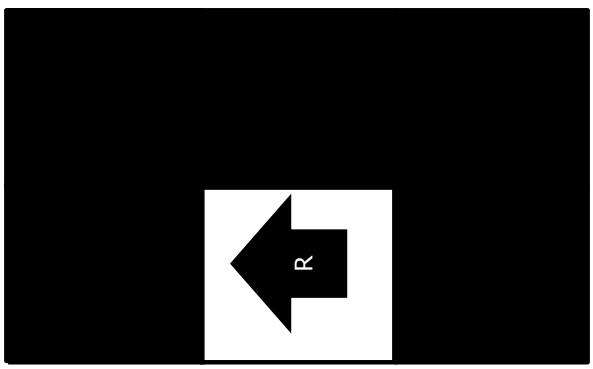
Input
FS = W

Hug The Wall: Case 3



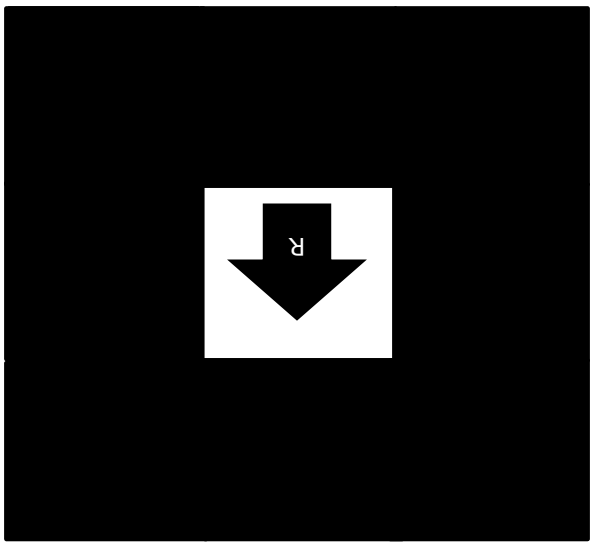
Output
Rotate left

Hug The Wall: Case 3

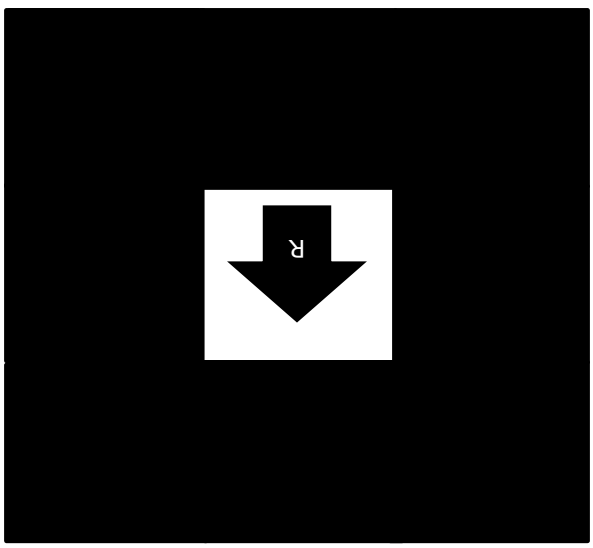


Input
FS = W

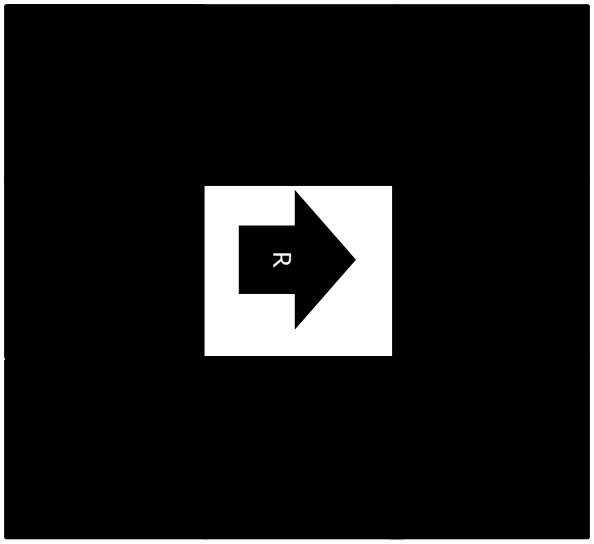
Hug The Wall: Case 3



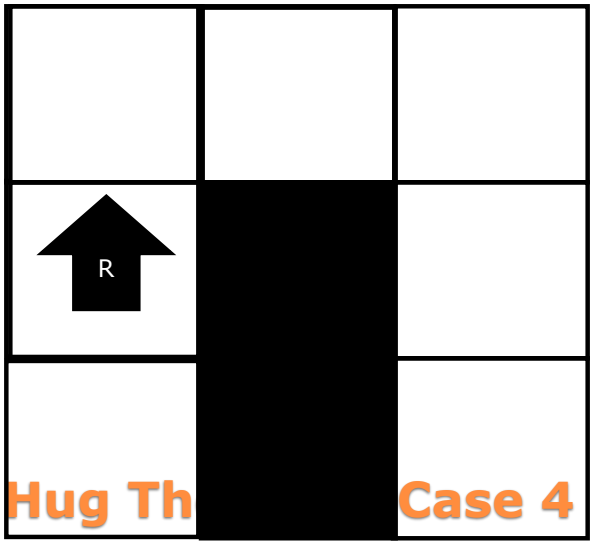
Output
Rotate left



Input
FS = W

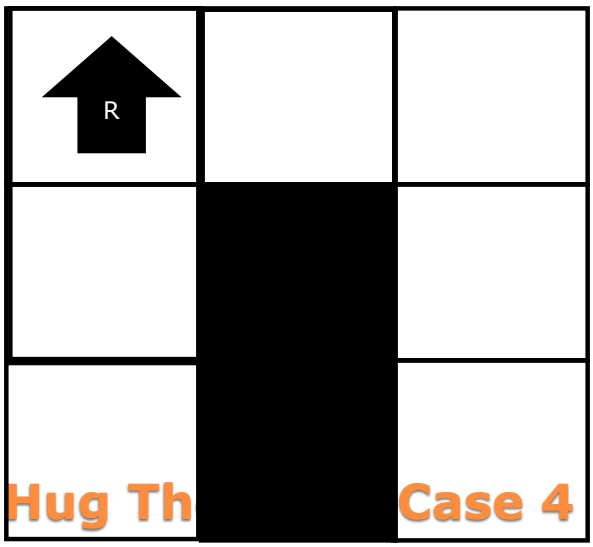


Output
Rotate left



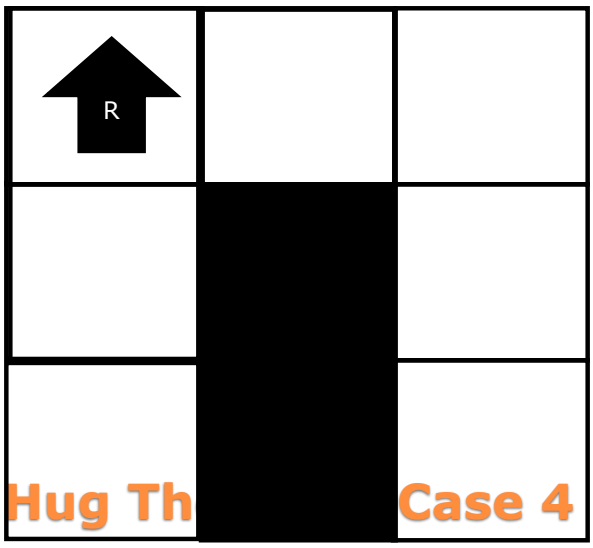
Input
FS = S
RS = W

Hug Th Case 4



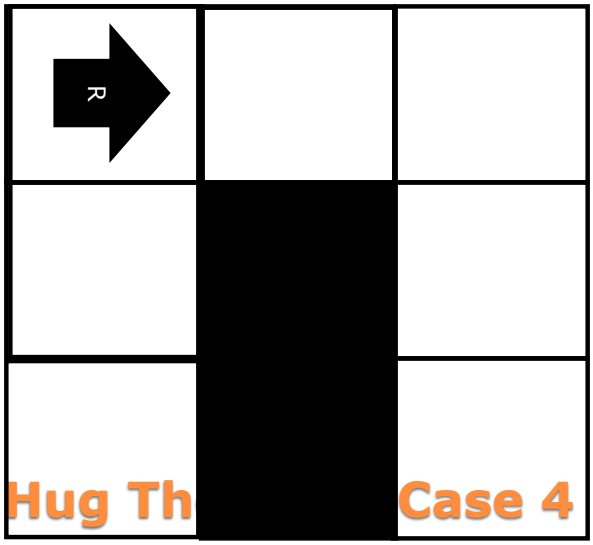
Output
Movement:
Forward

Hug Th Case 4



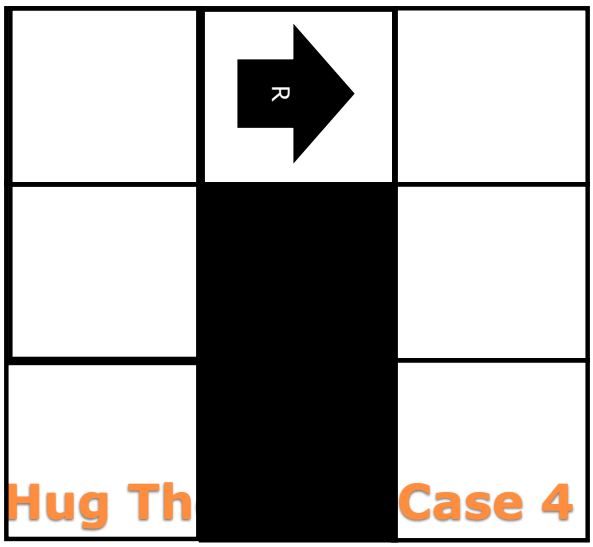
Input
FS = S
RS = S

Hug Th Case 4



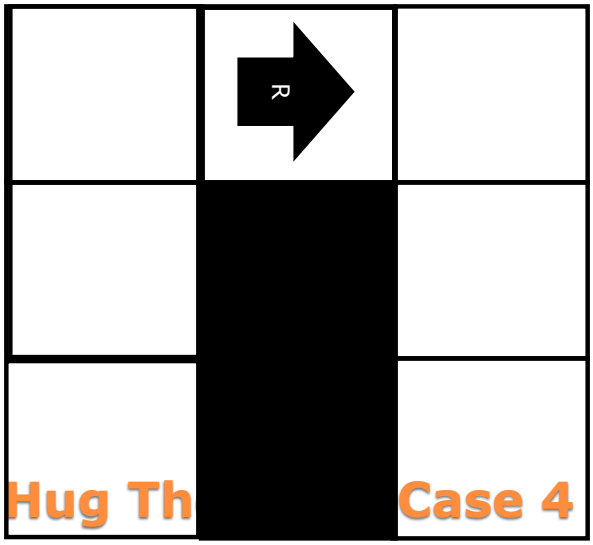
Output (Step 1)
Rotate: right

Hug Th Case 4



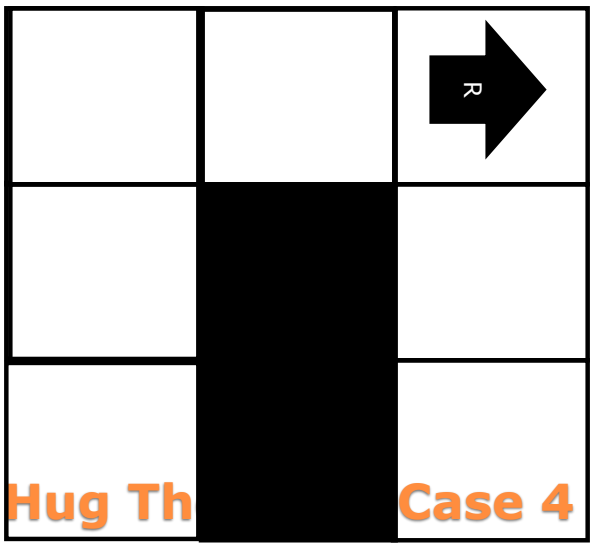
Output (Step 2):
Movement:
forward

Hug Th Case 4



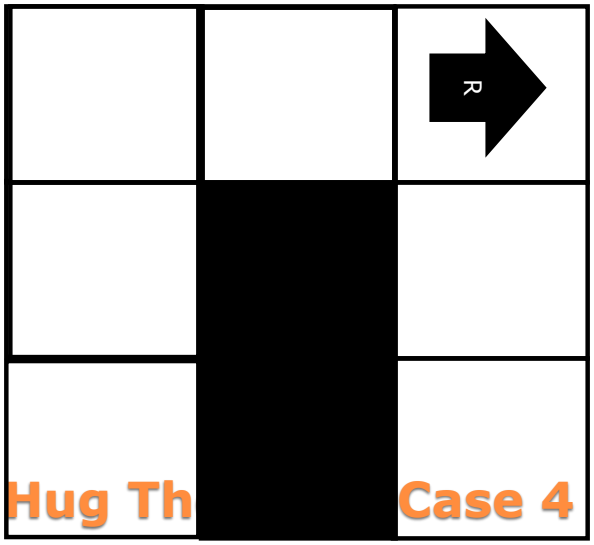
Input:
FS = S
RS = W

Hug Th Case 4



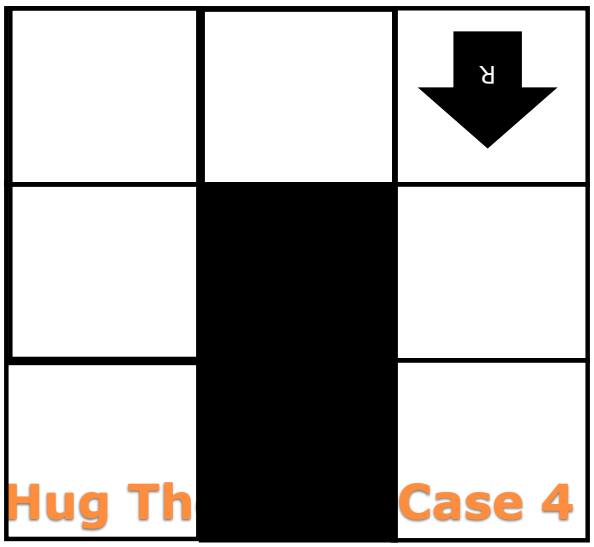
Output:
Movement:
forward

Hug Th Case 4



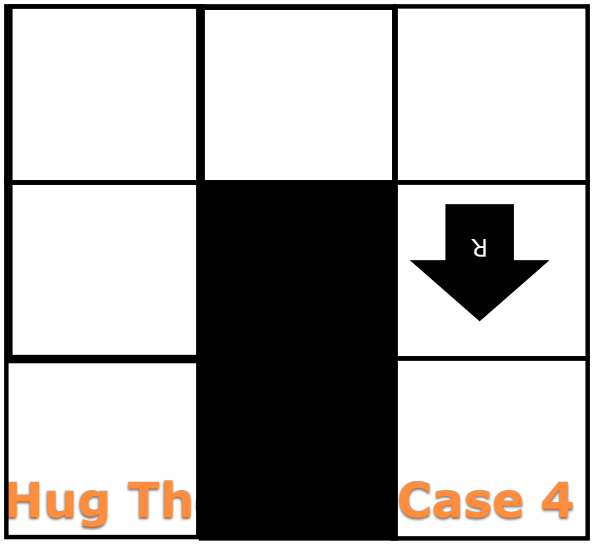
Input:
FS = S
RS = S

Hug Th Case 4



Output (step 1):
Rotate: right

Hug Th Case 4



Output (step 2):
Movement: foward

Hug Th Case 4

Repeat the following steps:

1. If **RS = W** and **FS = S**, then **F**
2. If **FS = W**, then **L**
3. If **RS = S** and **FS = S**, then **R** and **F**

Algorithm: Hug The Wall

