

# วิชาปัญญาประดิษฐ์ (Artificial Intelligence)

อ. พิชัย จอดพิมาย

Email : [pichaiku@gmail.com](mailto:pichaiku@gmail.com)

## แผนการสอน

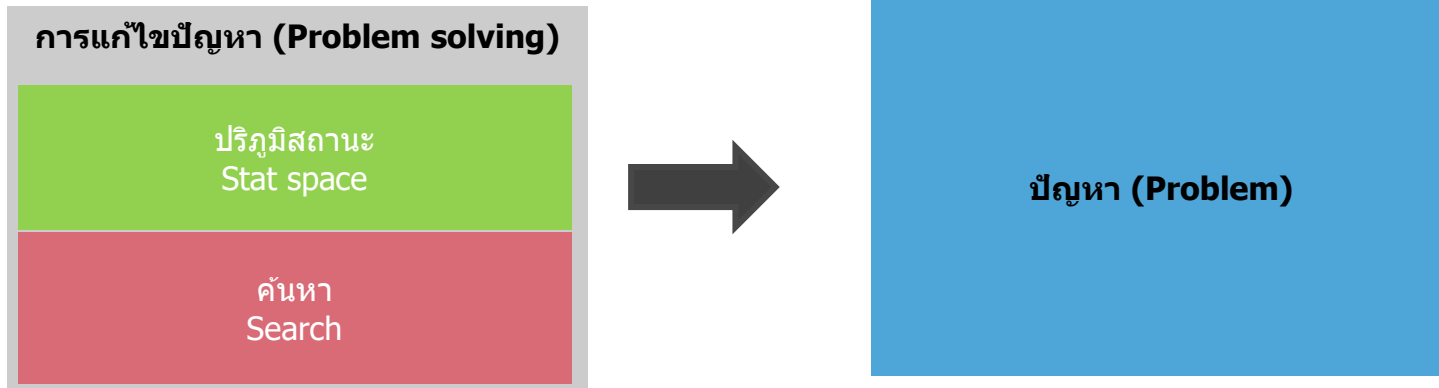
- ❑ สัปดาห์ที่ 1 : นิยาม ความสำคัญ และเทคโนโลยีของปัญญาประดิษฐ์
- ❑ สัปดาห์ที่ 2-3 : ปริภูมิสถานะและการค้นหา
- ❑ สัปดาห์ที่ 4-5 : ตรรกศาสตร์ประพจน์และตรรกศาสตร์พรีดิเคต
- ❑ สัปดาห์ที่ 6-8 : การประยุกต์ใช้ในเกม หุ่นยนต์ และระบบผู้เชี่ยวชาญ
- ❑ สัปดาห์ที่ 9 : **สอบกลางภาค**
- ❑ สัปดาห์ที่ 10-11 : การประมวลผลภาษาธรรมชาติ
- ❑ สัปดาห์ที่ 12-13 : การประมวลผลภาพ
- ❑ สัปดาห์ที่ 14-15 : การเรียนรู้ของเครื่องจักร
- ❑ สัปดาห์ที่ 16-18 : การประยุกต์ใช้ในเหมืองข้อมูลและการวิเคราะห์ข้อมูลขนาดใหญ่
- ❑ สัปดาห์ที่ 19 : **สอบปลายภาค**

# ปริภูมิสถานะและการค้นหา (State space and search)



# นิยามปริภูมิสถานะและการค้นหา

- ❑ ปริภูมิสถานะ (state space/search space) คือ ชุดของคำตอบหรือแนวทางการแก้ไขปัญหาที่เป็นไปได้
- ❑ การค้นหา คือ การค้นหาคำตอบจากชุดของคำตอบที่เป็นไปได้



# การกำหนดปัญหา (Problem representation)

- ❑ กำหนดสถานะเริ่มต้น (Initial state) คือ การกำหนดจุดเริ่มต้นในการค้นหาคำตอบ
- ❑ กำหนดสถานะเป้าหมาย (Goal state) คือ การกำหนดเป้าหมายหรือคำตอบ
- ❑ กำหนดตัวกระทำ (Operator) คือ การกำหนดชุดของวิธีการในการเปลี่ยนสถานะจากสถานะหนึ่ง ไปยังอีกสถานะหนึ่ง
- ❑ กำหนดค่าใช้จ่าย (Cost) คือ การกำหนดค่าใช้จ่าย เวลา หรือทรัพยากรที่ตัวกระทำดำเนินการไปในแต่ละครั้ง

## ตัวอย่างการกำหนดปัญหา จากเกมส์ 8 Puzzle

	5	2
1	8	4
7	6	3

สถานะเริ่มต้น

ให้ตัวกระทำ คือ Up (U),  
Down (D), Left (L) และ Right (R)



\*ระหว่างสถานะเริ่มต้นไปยัง  
เป้าหมาย จะเกิดสถานะต่าง ๆ  
มากมาย

1	5	2
7	8	4
6		3

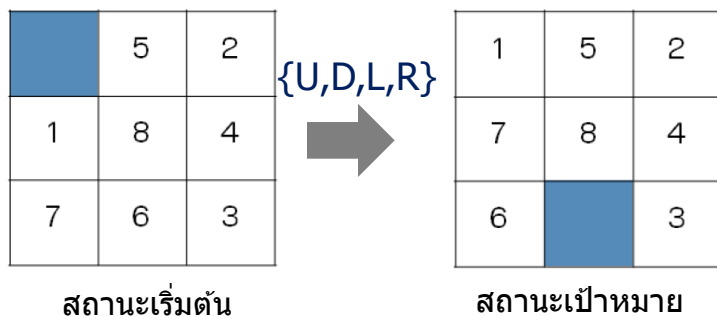
สถานะเป้าหมาย

*รูปแสดงสถานะเริ่มต้นและเป้าหมายของเกมส์ Puzzle*

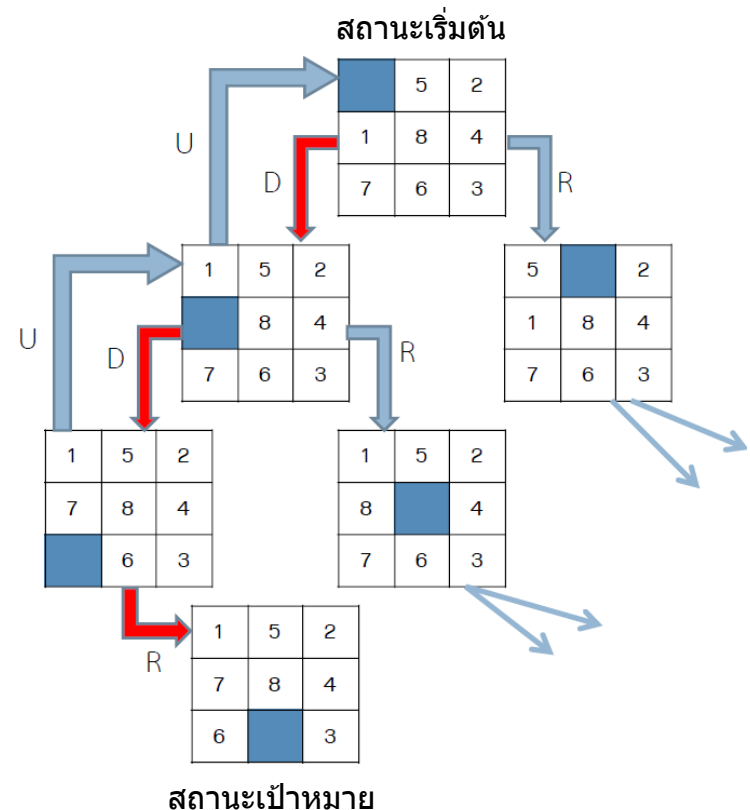
# การแก้ไขปัญห (Problem solving)

- ❑ สร้างปริภูมิสถานะ (ตัวเลือกหรือทางเลือก)
- ❑ ทำการค้นหา

ตัวอย่างแก้ไขปัญห จากเกมส์ 8 Puzzle



รูปแสดงสถานะเริ่มต้นและเป้าหมายของ  
เกมส์ Puzzle



รูปแสดงการกำหนดปริภูมิสถานะและการ  
ค้นหาเป้าหมายของเกมส์ Puzzle

# เทคนิคการค้นหา (Search technique)

- ❑ การค้นหาแบบบอด (Blind search)

เป็นเทคนิคที่ไม่มีตัวช่วยในการค้นหา ประกอบด้วย

- ❑ การค้นหาแนวกว้างก่อน (Breadth-first search)
- ❑ การค้นหาแนวลึกก่อน (Depth-first search)

- ❑ การค้นหาแบบฮิวริสติก (Heuristic search)

เป็นเทคนิคที่มีตัวช่วยในการค้นหา หรือตัวช่วยในการเลือกเส้นทาง ประกอบด้วย

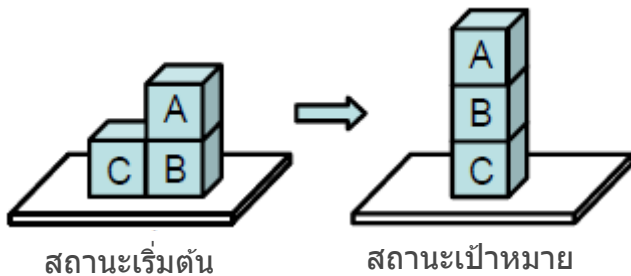
- ❑ การค้นหาแบบปีนเขา (Hill-climbing search)
- ❑ การค้นหาแบบละโมภ (Greedy search)
- ❑ การค้นหาแบบดีที่สุดก่อน (Best-first search)

# การค้นหาแนวกว้างก่อน (Breadth-first search)

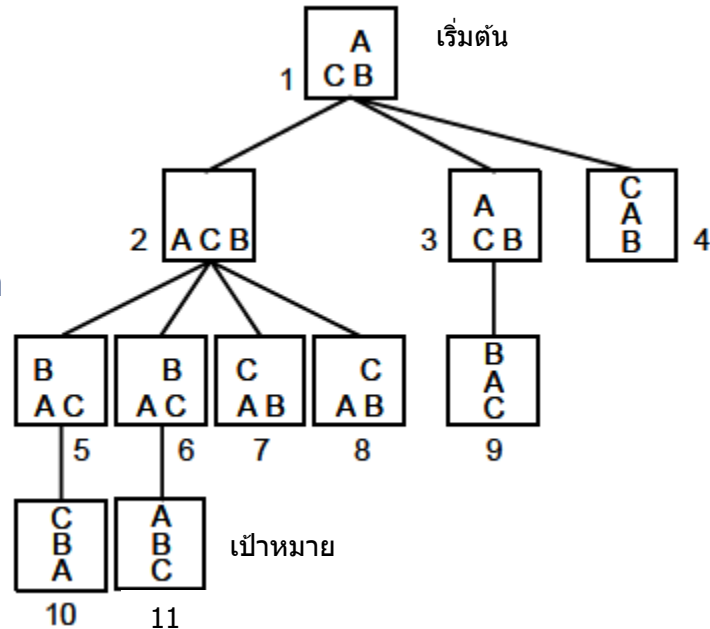
เป็นรูปแบบหนึ่งของการค้นหาแบบบอดโดยไม่มีตัวช่วยนำทางในการค้นหาคำตอบ  
วิธีการ คือ สร้างสถานะลูกให้ครบทุกสถานะ แล้วทำการตรวจสอบ ถ้าพบสถานะ  
เป้าหมายให้หยุด ถ้าไม่พบให้สร้างสถานะลูกต่อไป ทำในลักษณะนี้จนกว่าจะพบสถานะ  
เป้าหมาย

## ตัวอย่างแก้ไขปัญหการจัดเรียงกล่อง (World block problem)

ตัวกระทำ คือ การวางกล่องโดย  
ไม่สนใจว่าจะเลือกวางกล่องไหนก่อน  
ไม่สนใจตำแหน่งซ้ายขวา (ถือเป็นตำแหน่งเดียวกัน)  
ไม่สร้างสถานะใหม่ถ้าตัวเลือกเหมือนกับสถานะก่อนหน้า



รูปแสดงสถานะเริ่มต้นและเป้าหมายของ  
ปัญหการจัดเรียงกล่อง

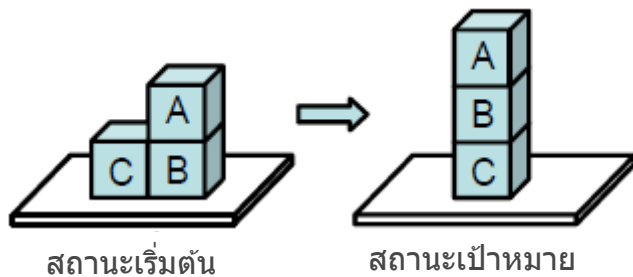


รูปแสดงการกำหนดปริภูมิสถานะและการค้นหา  
เป้าหมายของปัญหการจัดเรียงกล่อง

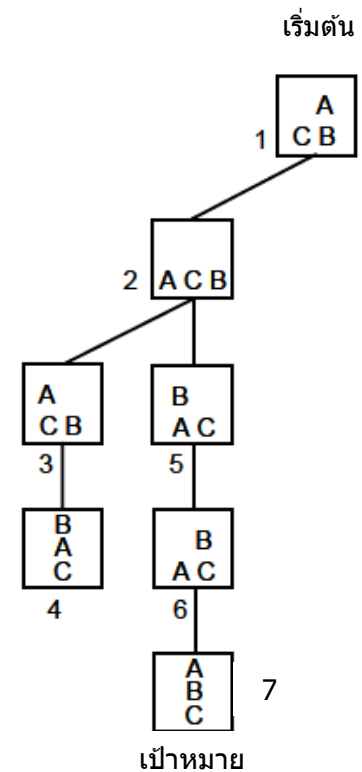


# การค้นหาแบบลึกก่อน (Depth-first search)

เป็นรูปแบบหนึ่งของการค้นหาแบบบอด โดยไม่มีตัวช่วยนำทางในการค้นหาคำตอบ  
วิธีการ คือ สร้างสถานะลูกมาหนึ่งสถานะ แล้วตรวจสอบ ถ้าพบสถานะเป้าหมายให้หยุด  
ถ้าไม่พบให้สร้างสถานะลูกต่อ ทำเช่นนี้จนกว่าจะไม่สามารถสร้างสถานะลูกได้ จากนั้น  
ค่อยย้อนกลับมาสร้างสถานะลูกต่อ ณ จุดที่สามารถสร้างสถานะลูกได้

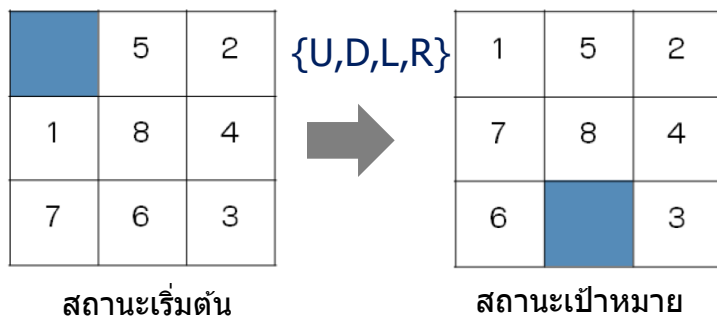


รูปแสดงสถานะเริ่มต้นและเป้าหมายของ  
ปัญหาการจัดเรียงกลอง

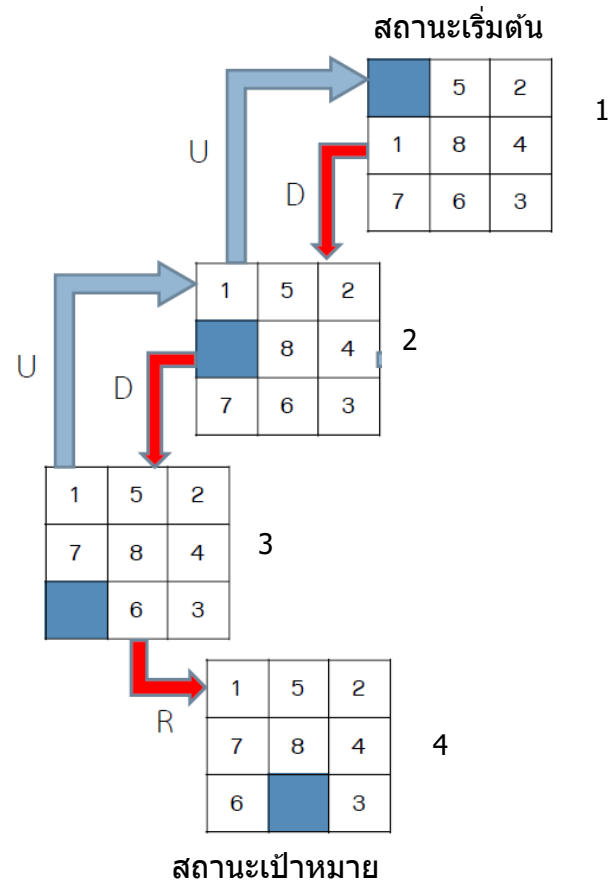


รูปแสดงการกำหนดปริภูมิสถานะและการค้นหา  
เป้าหมายของปัญหาการจัดเรียงกลอง

## ตัวอย่างการแก้ไขปัญหเกม 8 puzzle



รูปแสดงสถานะเริ่มต้นและเป้าหมายของ  
เกมส์ Puzzle



รูปแสดงการกำหนดปริภูมิสถานะและการ  
ค้นหาเป้าหมายของเกมส์ Puzzle

# การค้นหาแบบปีนเขา (Hill-climbing search)

เป็นรูปแบบหนึ่งของการค้นหาแบบฮิวริสติกโดยใช้ค่าฮิวริสติกเป็นแนวทางในการค้นหาคำตอบ **วิธีการ** คือ สร้างสถานะลูก (ทางเลือก) มา 1 สถานะ ถ้าค่าฮิวริสติกดีกว่าค่าฮิวริสติกของสถานะแม่ ให้ทำการสร้างสถานะย่อยต่อไปได้เลย ทำเช่นนี้จนกว่าจะพบสถานะเป้าหมาย (คำตอบ)

**ค่าฮิวริสติก** = ผลรวมของตำแหน่งที่เป็นไปตามกฎ

**กฎ** คือ

เทียบตำแหน่งปัจจุบันกับตำแหน่งเป้าหมาย

ถ้าตรงกันให้ + ตามจำนวนของกล่องที่เป็นฐาน

ถ้าไม่ตรงกันให้ - ตามจำนวนของกล่องที่เป็นฐาน

ถ้าไม่ทับกล่องอื่นเลยให้ 0

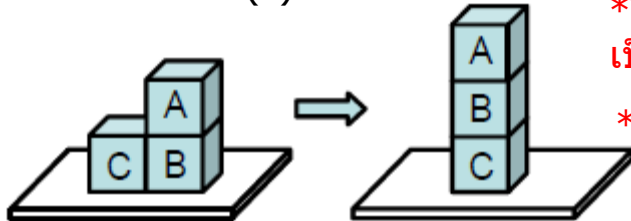
จากสถานะ (1) จะได้ว่า

A=-1 เนื่องจาก ไม่อยู่บน B และ C

B= 0 เนื่องจาก ไม่ทับกล่องอื่นเลย

C= 0 เนื่องจาก ไม่ทับกล่องอื่นเลย

ดังนั้นสถานะ (1) = A+B+C = -1



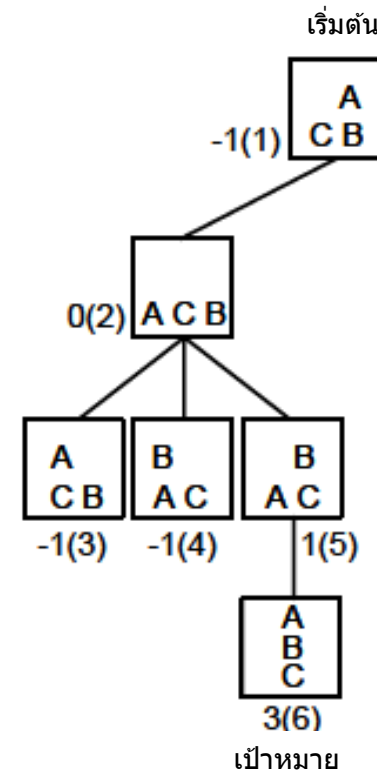
สถานะเริ่มต้น

สถานะเป้าหมาย

รูปแสดงสถานะเริ่มต้นและเป้าหมายของ  
ปัญหาการจัดเรียงกล่อง

\*ในกรณีนี้ใช้ค่าฮิวริสติก  
เป็นตัวกระทำ

\*ค่าฮิวริสติกยิ่งมากยิ่งดี



รูปแสดงการกำหนดปริภูมิสถานะและการค้นหา  
เป้าหมายของปัญหาการจัดเรียงกล่อง

# การค้นหาแบบละโมภ (Greedy search)

เป็นรูปแบบหนึ่งของการค้นหาแบบฮิวริสติก โดยจะสร้างสถานะลูกทุกตัวแล้วเลือกสถานะที่ดีที่สุด ซึ่งจะทำให้มีโอกาสพบคำตอบมากกว่าแบบป็นเขาแต่ประสิทธิภาพต่ำกว่า **วิธีการ** คือ สร้างสถานะลูก (ทางเลือก) ทุกสถานะ แล้วเลือกสถานะ (ทางเลือก) ที่ดีที่สุดและดีกว่าสถานะแม่ (โดยพิจารณาจากค่าฮิวริสติก) แล้วทำการสร้างสถานะลูกต่อ ทำเช่นนี้จนกว่าจะพบสถานะเป้าหมาย (คำตอบ)

ค่าฮิวริสติก = ผลรวมของตำแหน่งที่เป็นไปตามกฎ

กฎ คือ

เทียบตำแหน่งปัจจุบันกับตำแหน่งเป้าหมาย

ถ้าตรงกันให้ + ตามจำนวนของกล่องที่เป็นฐาน

ถ้าไม่ตรงกันให้ - ตามจำนวนของกล่องที่เป็นฐาน

ถ้าไม่ทับกล่องอื่นเลยให้ 0

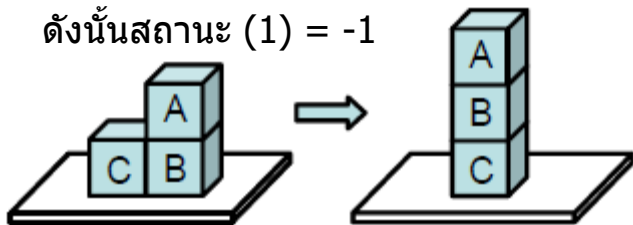
จากสถานะ (1) จะได้ว่า

A=-1 เนื่องจาก ไม่อยู่บน B และ C

B= 0 เนื่องจาก ไม่ทับกล่องอื่นเลย

C= 0 เนื่องจาก ไม่ทับกล่องอื่นเลย

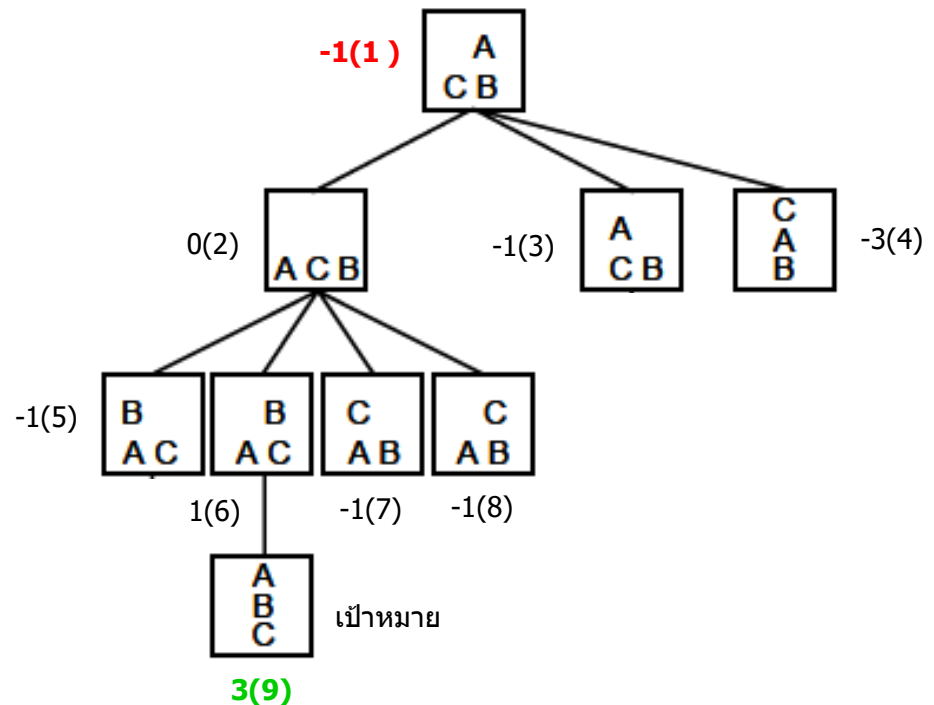
ดังนั้นสถานะ (1) = -1



สถานะเริ่มต้น

สถานะเป้าหมาย

รูปแสดงสถานะเริ่มต้นและเป้าหมายของ  
ปัญหาการจัดเรียงกล่อง



รูปแสดงการกำหนดปริภูมิสถานะและการค้นหา  
เป้าหมายของปัญหาการจัดเรียงกล่อง

## ตัวอย่างการแก้ไขปัญหาเกม 8 puzzle

ตัวอย่างของฟังก์ชันฮิวริสติกสำหรับปัญหา 8-Puzzle แสดงในสมการด้านล่างนี้

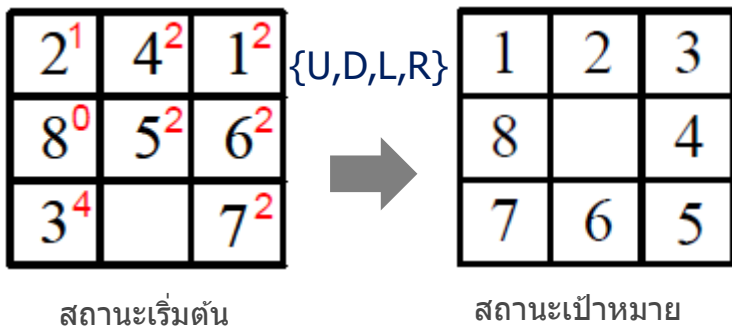
$$h_{\text{Man}} = \sum_{i=1}^8 d_x(c_i, g_i) + \sum_{i=1}^8 d_y(c_i, g_i)$$

โดยที่  $c_i$ ,  $g_i$ ,  $d_x$ ,  $d_y$  คือพิกัดของแผ่นป้าย  $i$  ที่สถานะปัจจุบัน พิกัดของแผ่นป้าย  $i$  ที่สถานะเป้าหมาย ระยะห่างระหว่าง  $c_i$  กับ  $g_i$  ตามแกน  $x$  และระยะห่างระหว่าง  $c_i$  กับ  $g_i$  ตามแกน  $y$  ตามลำดับ

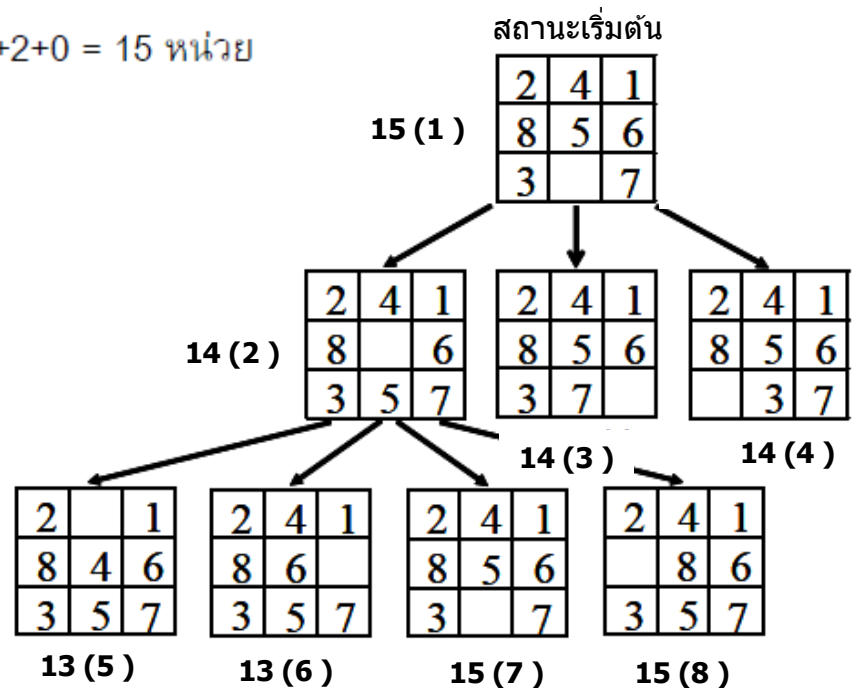
ฟังก์ชันนี้เรียกว่าฟังก์ชันแมนฮัตตัน ฟังก์ชันนี้สามารถแปลความหมายอย่างง่ายคือ การหาค่าจำนวนครั้งที่ต้องขยับแผ่นป้ายตั้งแต่แผ่นที่ 1 ถึงแผ่นที่ 8 จากตำแหน่งปัจจุบันตามแนวแกน  $x$  และ  $y$  ไปยังตำแหน่งที่ควรอยู่ตามคำตอบ ว่าต้องขยับทั้งสิ้นน้อยที่สุดรวมกี่ครั้ง โดยไม่ต้องมีเงื่อนไขว่าจะติดกับช่องว่างหรือไม่ (สามารถเลื่อนได้เลย แม้จะมีแผ่นป้ายอื่นวางอยู่แล้ว) สังเกตว่าฟังก์ชันนี้ยังมีค่าน้อยยิ่งดี

ตัวอย่างเช่นพิจารณารูปที่ 2-12 ตัวเลขที่อยู่มุมขวาบนในแผ่นป้ายแต่ละแผ่นแสดงจำนวนครั้งที่ต้องเลื่อนแผ่นป้ายนั้นไปยังตำแหน่งที่มันควรอยู่เมื่อเทียบกับคำตอบ เช่นแผ่นป้าย '2' ที่สถานะปัจจุบันต้องเลื่อนทั้งหมด 1 ครั้ง (ตามแนวแกน x) จึงจะไปอยู่ในตำแหน่งเดียวกับในคำตอบ แผ่นป้าย '3' ที่สถานะปัจจุบันต้องเลื่อนทั้งหมด 2+2 ครั้ง (2 ครั้งตามแนวแกน x และ 2 ครั้งตามแนวแกน y) จึงจะไปอยู่ในตำแหน่งเดียวกับในคำตอบ ดังนั้นค่าฮิวริสติกของสถานะปัจจุบันนี้มีค่าเท่ากับ  $2+1+4+2+2+2+2+0 = 15$  หน่วย

\*ค่าฮิวริสติกยิ่งน้อยยิ่งดี



รูปแสดงสถานะเริ่มต้นและเป้าหมายของ  
เกมส์ Puzzle



รูปแสดงการกำหนดปริภูมิสถานะและการ  
ค้นหาเป้าหมายของเกมส์ Puzzle

## ตัวอย่างการประเมินราคาอสังหาริมทรัพย์

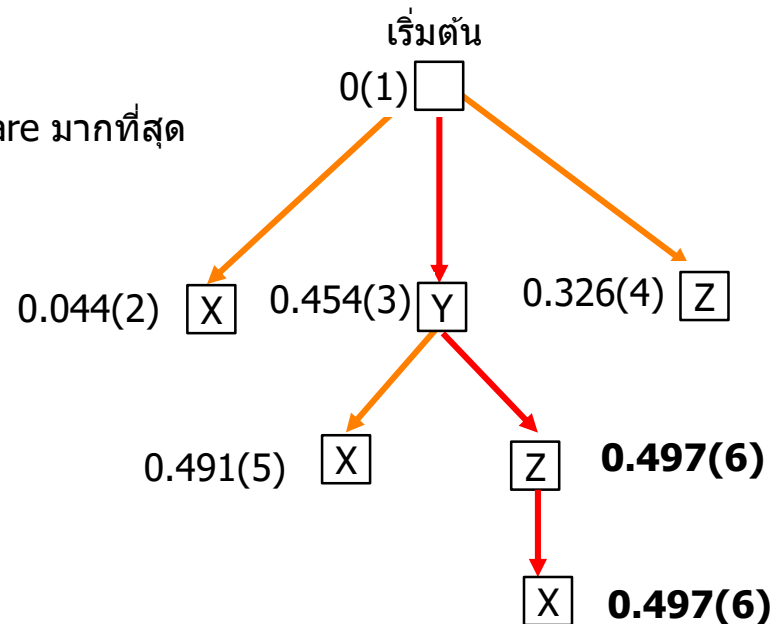
ลำดับ	x=อายุของบ้าน	y=ระยะทางไปยัง สถานีรถไฟฟ้า	z=จำนวนร้าน สะดวกซื้อที่อยู่ ใกล้เคียง	p=ราคาบ้านต่อ พื้นที่หน่วย (10,000 New Taiwan Dollar/3.3 ตร.ม.)
1	32	84.87882	10	37.9
2	19.5	306.5947	9	42.2
3	13.3	561.9845	5	47.3
4	13.3	561.9845	5	54.8
5	5	390.5684	5	43.1
6	7.1	2175.03	3	32.1
7	34.5	623.4731	7	40.3
8	20.3	287.6025	6	46.7
9	31.7	5512.038	1	18.8
10	17.9	1783.18	3	22.1
11	34.8	405.2134	1	41.4
12	6.3	90.45606	9	58.1

รูปแสดงตัวอย่างข้อมูล (ข้อมูลทั้งหมดอยู่ในไฟล์ AI-week 02-  
การประเมินราคาอสังหา-st.xlsx

สถานะเริ่มต้น = ไม่มีปัจจัยใด ๆ เลย  
สถานะเป้าหมาย = ชุดของปัจจัยที่มีค่า R square มากที่สุด  
ค่าฮิวริสติก = ค่า R square ที่ได้จาก  
Regression analysis

\* ให้ใช้ข้อมูลในไฟล์ที่เตรียมไว้ให้เพื่อทำ  
การคำนวณค่า R square

\* ค่าฮิวริสติกยิ่งมากยิ่งดี

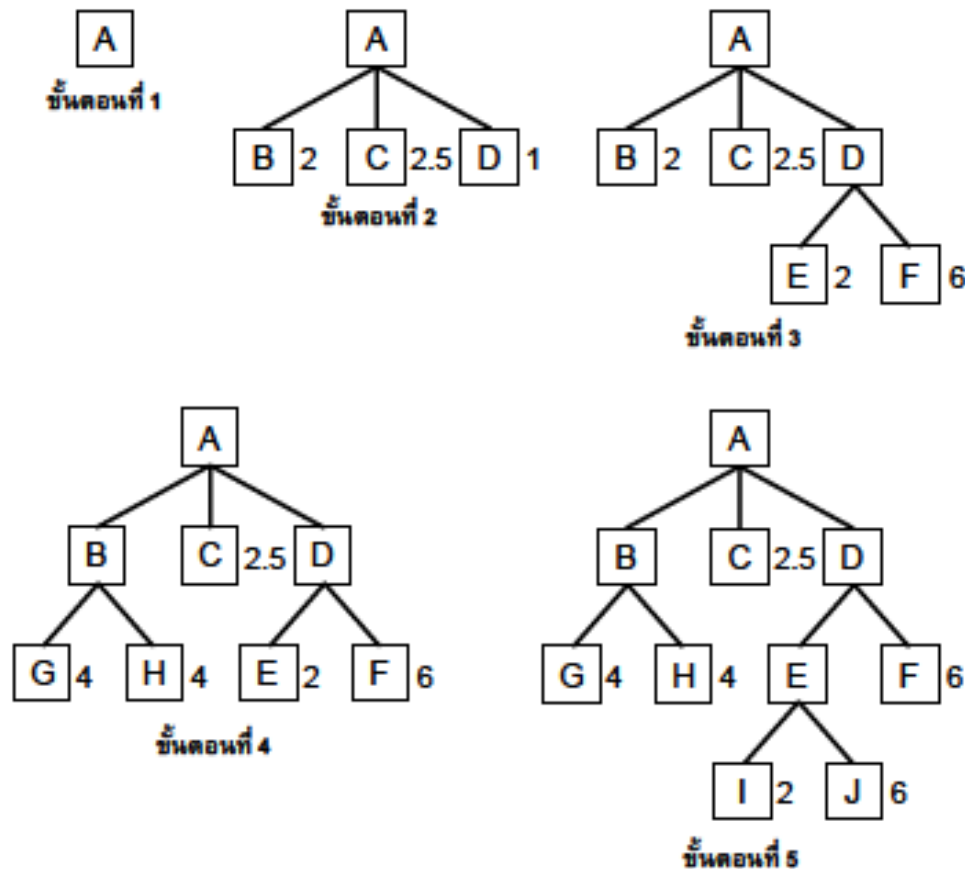


ดังนั้น  
เป้าหมาย = ปัจจัย Y, Z และ X



# การค้นหาแบบดีที่สุดก่อน (Best-first search)

เป็นรูปแบบหนึ่งของการค้นหาแบบฮิวริสติก จะเก็บสถานะทุกตัวโดยไม่มีการตัดทิ้ง ต่างจากอัลกอริทึมละโมภที่เมื่อเลือกเส้นทางหนึ่งแล้ว ตัวเลือกอื่นที่เป็นลูกของสถานะปัจจุบันจะถูกตัดทิ้งไป ซึ่งทำให้ไม่พลาดเส้นทางที่นำไปสู่คำตอบ



\*ค่าฮิวริสติกยิ่งน้อยยิ่งดี

รูปแสดงการกำหนดปริภูมิสถานะและการค้นหา  
เป้าหมายของปัญหาการจัดเรียงกล่อง

# เปรียบเทียบประสิทธิภาพและประสิทธิผล

เปรียบเทียบระหว่าง Blind search และ Heuristic search

- ☐ Blind search มักจะมีประสิทธิผล (ความถูกต้อง) มากกว่า Heuristic search
- ☐ Heuristic search มักจะมีประสิทธิภาพ (เวลา ต้นทุน จำนวนการเคลื่อนที่) Blind search

เปรียบเทียบระหว่าง Breadth-first และ Depth-first search

- ☐ Breadth-first และ Depth-first search มีประสิทธิผลและประสิทธิภาพเท่า ๆ กัน  
ทั้งนี้ขึ้นอยู่กับตำแหน่งของสถานะเป้าหมายหรือคำตอบ

เปรียบเทียบระหว่าง Hill-climbing, greedy, และ best-first search

- ☐ ประสิทธิภาพมักจะเรียงจาก Hill-climbing > Greedy > Best-first search
- ☐ ประสิทธิผลมักจะเรียงจาก Best-first > Greedy > Hill-climbing search

# แบบฝึกหัด

สำหรับข้อ 1-5 กำหนดให้สถานะเริ่มต้นและสถานะเป้าหมายของเกม puzzle เป็นดังนี้

1	2	3
8	5	6
4	7	

สถานะเริ่มต้น

1	2	3
8	6	
4	5	7

สถานะเป้าหมาย

โดยให้ ตัวกระทำการ={U,D,L,R}

สำหรับข้อ 6-7 ให้ใช้ข้อมูลในฐานข้อมูล UCI

<https://archive.ics.uci.edu/ml/datasets.html> ในการหาปัจจัยที่สำคัญ

1. ให้ใช้เทคนิค Breadth-first search เพื่อการค้นหาสถานะเป้าหมาย
2. ให้ใช้เทคนิค Depth-first search เพื่อการค้นหาสถานะเป้าหมาย
3. ให้ใช้เทคนิค Hill-climbing search เพื่อการค้นหาสถานะเป้าหมาย
4. ให้ใช้เทคนิค Greedy search เพื่อการค้นหาสถานะเป้าหมาย
5. ให้ใช้เทคนิค Best-first search เพื่อการค้นหาสถานะเป้าหมาย
6. ให้ใช้เทคนิค Breadth-first search เพื่อการหาชุดของปัจจัยสำคัญต่อการทำนาย
7. ให้ใช้เทคนิค Greedy search เพื่อการหาชุดของปัจจัยที่สำคัญต่อการทำนาย