Lab-5 Question-2

2 dimensional Array Map

$$\begin{pmatrix}
-1 & -9 & -1 & -1 & -1 & -9 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -9 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -9 & -1 & -1 & -9 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -9 & -1 \\
-1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1 & -1 \\
-9 & -1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1 \\
-1 & -9 & -1 & -1 & -1 & -9 & -1 & -1 & -9 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1
\end{pmatrix}$$

กำหนดจุดเริ่มต้นและจุดสุดท้ายคือ α (0,0) และ β (0,9) ตามลำดับ

ผลลัพธที่ได้ของการใช้ FindPath ที่ใช้ Queue จะเท่ากับ

FindPath using Queue

$$\begin{pmatrix}
0 & -9 & 4 & 5 & 6 & -9 & 10 & 11 & 12 & 13 \\
1 & 2 & 3 & 4 & -9 & 8 & 9 & 10 & 11 & 12 \\
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\
3 & 4 & 5 & 6 & -9 & 8 & 9 & -9 & 11 & 12 \\
4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & -9 & 13 \\
5 & 6 & 7 & 8 & 9 & -9 & 11 & 12 & 13 & 14 \\
-9 & 7 & 8 & 9 & 10 & 11 & -9 & 13 & 14 & -1 \\
9 & 8 & 9 & 10 & 11 & 12 & -9 & 14 & -1 & -1 \\
10 & -9 & 10 & 11 & 12 & -9 & -1 & -1 & -9 & -1 \\
10 & -9 & 10 & 11 & 12 & -9 & -1 & -1 & -9 & -1
\end{pmatrix}$$

71 Step

ผลลัพธที่ได้ของการใช้ FindPath ที่ใช้ Stack จะเท่ากับ

FindPath using Stack

$$\begin{pmatrix}
0 & -9 & 4 & 5 & 6 & -9 & -1 & -1 & -1 & 13 \\
1 & 2 & 3 & 4 & -9 & 8 & 9 & 10 & 11 & 12 \\
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\
-1 & -1 & -1 & 6 & -9 & 8 & 9 & -9 & 11 & 12 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -9 & -1 \\
-1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1 & -1 \\
-9 & -1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1 \\
-1 & -9 & -1 & -1 & -1 & -9 & -1 & -1 & -9 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1
\end{pmatrix}$$

15 Step

หากมองแบบนี้จะเห็นได้ว่า หากใช้ขั้นตอนวิธีเดิมในการหาวิถีที่สั้นที่สุดแต่เปลี่ยน Data structure จาก Queue เป็น Stack ขั้นตอนวิธีนี้จะเร็วขึ้นอย่างเห็นได้ชัด จากการวนซ้ำ 71 ครั้งลดลงเหลือแค่ 15 ครั้ง แต่หากเราลองวิเคราะห์ดีๆจะพบว่าที่ขั้นตอนวิธีนี้เร็วขึ้นเมื่อเปลี่ยน Queue เป็น Stack นั้นเป็นเพราะว่าเราหาวิถีจากจุดซ้ายบนสุด (0,0) ไปทาง **ขวาบนสุด** (0,9) เมื่อเราแทนที่ Queue ด้วย Stack ในแต่ละขั้นที่จะมีการเติม $k+1$ ที่ช่องข้างๆ k เมื่อเติมเสร็จแล้วจะนำช่องที่เติมไปใส่ไว้ใน Stack และช่อง $k+1$ ที่โดนเติมล่าสุดจะอยู่ในชั้นบนสุดของ Stack และเมื่อมีการวนซ้ำครั้งถัดไปก็จะนำช่องนี้มาหาช่องรอบๆและช่องรอบๆที่เป็นช่องล่าสุดที่โดนเติมก็จะเป็นตัวถัดไปที่จะนำไปเติมช่องถัดๆไปซึ่งทำให้ช่องเก่าๆบางช่องนั้นไม่ได้ถูกนำไปพิจารณาสักที และการเติมจะมีรูปแบบที่เติมไปเรื่อยๆเข้าหาทิศทางๆหนึ่งเป็นพิเศษ ทิศทางนั้นก็คือทิศที่จะเติมเป็นทิศล่าสุดของการวนซ้ำในแต่ละรอบหากทิศนั้นไม่เจอกับกำแพง สำหรับ Source code ที่ให้มานั้นจะมีการใช้ expand(); ไปในทางบน ล่าง ขวา ซ้าย ตามลำดับแต่เนื่องจากจุดเริ่มต้นคือจุด (0,0) ทำให้การ expand ไปทางซ้ายนั้นติดกำแพง expand ตัวที่ทำล่าสุดที่เดินได้จึงเป็นการ expand ไปทางขวา ในการวนครั้งถัดไปฝั่งซ้ายซึ่งเป็นตัวก่อนหน้าก็ติดทำให้ expand ล่าสุดที่เป็นไปได้ก็คือทางขวาไปเรื่อยๆจึงเห็นได้ว่า FindPath แบบใช้ Stack นี้จะไล่เติมเลขทาง**ขวา**ก่อน หากเราวางจุดสุดท้ายไว้ทางขวาของจุดเริ่มต้นจึงทำให้ FindPath Stack สามารถหาวิถีได้เร็วกว่า FindPath Queue

หากจุดสุดท้ายไม่ได้อยู่ทางขวาบนสุดแต่อยู่ที่จุดใดๆ ขั้นตอนวิธีนี้ ที่มีการใช้ FindPath Stack แทน Queue จะไม่สามารถทำงานได้อย่างถูกต้องเพราะอย่างที่กล่าวข้างต้นขั้นตอนวิธีนี้จะไล่เติมเลขไปทางขวาเรื่อยๆหากสุดกำแพงทางขวาแล้วทิศทางที่จะไล่เลขต่อมาก็จะลงมาล่างเล็กน้อยและต่อไปจนซ้ายสุดและเมื่อเป็นแบบนี้ทำให้ช่องแรกๆที่อยู่ในก้น Stack ต่อให้ถูกดึงออกไปประมวลผลก็ไม่สามารถเติมเลข $k+1$ ลงไปได้เนื่องจากช่องพวกนี้ถูกเติมไปหมดแล้วโดยการไล่เลขจากขวามาซ้ายที่กล่าวไปข้างต้น ทำให้เมื่อจุดสุดท้ายอยู่ข้างล่างของจุดเริ่มต้นขั้นตอนวิธีนี้จะคำนวณวิถีผิดและได้ผลดังนี้

กำหนดจุดเริ่มต้นและจุดสุดท้ายคือ $\alpha\ (0,0)$ และ $\beta\ (3,0)$ ตามลำดับ

ผลลัพธที่ได้ของการใช้ FindPath ที่ใช้ Queue จะเท่ากับ

FindPath using Queue

$$\begin{pmatrix} 0 & -9 & -1 & -1 & -1 & -9 & -1 & -1 & -1 & -1 \\ 1 & 2 & 3 & -1 & -9 & -1 & -1 & -1 & -1 & -1 \\ 2 & 3 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 3 & -1 & -1 & -1 & -9 & -1 & -1 & -9 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -9 & -1 \\ -1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1 & -1 \\ -9 & -1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1 \\ -1 & -9 & -1 & -1 & -1 & -9 & -1 & -1 & -9 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1 \end{pmatrix}$$

4 Step

ผลลัพธที่ได้ของการใช้ FindPath ที่ใช้ Stack จะเท่ากับ

FindPath using Stack

$$\begin{pmatrix} 0 & -9 & 4 & 5 & 6 & -9 & 16 & 15 & 14 & 13 \\ 1 & 2 & 3 & 4 & -9 & 8 & 9 & 10 & 11 & 12 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 25 & 24 & 23 & 6 & -9 & 8 & 9 & -9 & 11 & 12 \\ 24 & 23 & 22 & 21 & 20 & 19 & 18 & 17 & -9 & 13 \\ 25 & 24 & 23 & 22 & 21 & -9 & 17 & 16 & 15 & 14 \\ -9 & -1 & -1 & -1 & -1 & -1 & -9 & 17 & 16 & 15 \\ -1 & -1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1 \\ -1 & -9 & -1 & -1 & -1 & -9 & -1 & -1 & -9 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -9 & -1 & -1 & -1 \end{pmatrix}$$

32 Step

จะเห็นได้ว่านอกจากขั้นตอนในการรันของ FindPath Queue จะกลับมาเร็วกว่าแล้วผลรันของ FindPath Stack ยังออกมาผิดอีกด้วย โดยจุดเริ่มต้นคือ (0,0) จุดสุดท้ายคือ (3,0) และไม่มี Obstacle มาบังการเดินลงแบบตรงๆแต่วิถีที่ได้จาก FindPath Stack นั้นกลับมีการเดินอ้อมจากทางซ้ายไปขวาและวนกลับขวาไปซ้าย เนื่องจากการเก็บช่องลงใน Stack และดึงช่องล่าสุดออกมาใช้เรื่อยๆแบบนี้จึงทำให้ตำแหน่ง (2,0) ไม่ได้ถูกนำมาประมวลผลและเติม $k+1$ ให้กับ (3,0) หรือต่อให้ถูกนำมาประมวลผล แต่เนื่องจากการไล่เลขแบบที่กล่าวไปทำให้ตำแหน่ง (3,0) ถูกเติมด้วย $k+1$ ของช่องที่ถูกไล่มาจากทางขวาไปก่อนแล้วจึงทำให้ไม่ได้ระยะทางที่สั้นที่สุด

จากทั้งหมดที่กล่าวมาจึงขอตอบคำถาม

Q : ให้ใช้คลาส Stack แทนคลาส Queue ในการหาวิถี (path) แล้วดูว่า วิถี (path) ที่ได้มานี้แตกต่างจากเดิมอย่างไร เพราะเหตุใด

A : วิถี (path) ที่ได้จากการแทน Queue ด้วย Stack สำหรับ algorithm จาก source code นี้ในบางกรณีแล้วจะไม่ใช้วิถีที่สั้นที่สุด เนื่องจากเมื่อใช้ Stack เก็บช่องที่รอประมวลผลช่องที่เก็บล่าสุดจะถูกดึงมาทำงานเรื่อยๆจนทำให้เกิดการไล่เลขไปในทิศทางเดียวและในบางครั้งจะทำให้การไล่เลขไม่ได้ออกมาเป็นวิถีที่สั้นที่สุด หากจะทำ Dept First Search จะต้องเขียนขั้นตอนวิธีใหม่ไม่ใช่แค่แทน Queue ลงไปด้วย Stack