

โรคระบาด (ฉบับร่าง)

เราต้องการจำลองสถานการณ์การระบาดของโรคติดต่อในพื้นที่ขนาด $W \times H$ ตารางหน่วย พื้นที่ดังกล่าวมีลักษณะเป็นสี่เหลี่ยมผืนผ้า เพื่อความสะดวกในการพิจารณาต่อไป เราจะสมมติว่ามุมทั้งสี่ของพื้นที่ดังกล่าวอยู่ที่พิกัด $(0,0)$, $(W,0)$, $(0,H)$ และ (W,H) บนระนาบสองมิติ

ในพื้นที่ดังกล่าวมี

- ประชากรทั่วไปที่ไม่นับแพทย์จำนวน N คน ในกลุ่มนี้ บางคนอาจจะเป็นโรค
- แพทย์จำนวน D คน เราสมมติว่าแพทย์จะไม่มีทางติดโรค

ในการจำลองสถานการณ์ เราจะพิจารณาพฤติกรรมของสมาชิกในระบบเป็นรอบ ๆ โดยในแต่ละรอบ ประชากรธรรมดาจะเคลื่อนที่ จากนั้นจะมีการแพร่กระจายของโรคระหว่างประชากรที่อยู่ใกล้กัน จากนั้นแพทย์แต่ละคนจะรักษาผู้ป่วยที่อยู่ในรัศมีการรักษา และสุดท้ายแพทย์จะเดินทางไปยังตำแหน่งใหม่ เพื่อรักษาคนไข้ต่อไป

เมื่อเริ่มต้นระบบจะอยู่ที่เวลา 0 โปรแกรมจะจำลองพฤติกรรมเป็นรอบ ๆ สำหรับการจำลองที่เริ่มในเวลา t โปรแกรมจำลองพฤติกรรมของประชากรและแพทย์ตามลำดับต่อไปนี้ จากนั้นเมื่อจำลองเสร็จระบบจะเข้าสู่เวลา $t + 1$

1. ประชากร

ในแต่ละรอบ ประชากรแต่ละคนจะสุ่มทิศทางการเคลื่อนที่ จากนั้นจะย้ายตำแหน่งไปยังตำแหน่งใหม่โดยเคลื่อนที่ไปในทิศทางดังกล่าวเป็นระยะทาง 1 หน่วย (ถ้าไม่ออกนอกพื้นที่) จากนั้นสำหรับคนที่อยู่ใกล้กับผู้ป่วยในระยะห่างอย่างมาก C หน่วยจะมีโอกาสติดโรคด้วยความน่าจะเป็น p

ประชากรที่ป่วยเป็นโรคแล้วจะไม่หายจนกว่าแพทย์จะรักษา และเมื่อหายแล้ว สามารถติดโรคซ้ำได้อีก (ในโมเดลของเราจะไม่คำนึงถึงภูมิคุ้มกันหรือการดื้อยา)

2. แพทย์

แพทย์แต่ละคนจะมีรัศมีการรักษา R หน่วย นั่นคือจะรักษาคนป่วยที่อยู่ในระยะ R หน่วยจากตนเองเท่านั้น นอกจากนี้ แพทย์แต่ละคนจะสามารถรักษาคนไข้ได้ไม่เกิน H คนต่อหนึ่งช่วงเวลา (รายละเอียดว่าแพทย์จะรักษาใครจะระบุต่อไป)

เมื่อรักษาเสร็จแล้ว แพทย์จะเดินทางโดยแพทย์มีรถรับส่งทำให้เคลื่อนที่ไปได้ระยะทาง S ไม่เกินหน่วย (จะเคลื่อนที่ไปทิศทางใดจะระบุต่อไป)

ลำดับในการจำลองสถานการณ์ในแต่ละรอบเวลา จะเริ่มโดยจำลองการเคลื่อนที่ของประชากร จำลองการติดต่อกันของโรค จำลองการรักษาของแพทย์ และสุดท้ายจำลองการเดินทางของแพทย์ โดยรายละเอียดต่าง ๆ จะอธิบายเพิ่มเติมต่อไป

งานของคุณ

เขียนโปรแกรมรับพารามิเตอร์ของการจำลองต่าง ๆ (ดูสรุปได้จากตารางที่ 1) จากนั้นคำนวณหาตำแหน่งของประชากร รวมถึงสถานะว่าเป็นโรคหรือไม่ และตำแหน่งของแพทย์ในเวลาต่าง ๆ

ตารางที่ 1 สรุปพารามิเตอร์ของการจำลองสถานการณ์

ตัวแปร	ความหมาย
W, H	ขนาดของพื้นที่ (กว้าง, สูง)
N	จำนวนประชากรธรรมดา (ไม่นับรวมแพทย์)
D	จำนวนแพทย์
C	ระยะติดโรค
p	ความน่าจะเป็นที่จะติดโรค
R	ระยะที่แพทย์สามารถรักษาโรคได้
H	จำนวนคนมากที่สุดที่แพทย์หนึ่งคนสามารถรักษาได้ในแต่ละช่วงเวลา
S	ระยะทางมากที่สุดที่แพทย์สามารถเคลื่อนที่ได้

ข้อมูลนำเข้า

อ่านข้อมูลเหล่านี้จากแฟ้ม pandemic.in

- บรรทัดที่ 1 ระบุจำนวนเต็มสองจำนวน W และ H ($1 \leq W \leq 10,000$; $1 \leq H \leq 10,000$)
- บรรทัดที่ 2 ระบุจำนวนเต็มสองจำนวน N และ D ($1 \leq N \leq 1,000,000$; $1 \leq D \leq 100,000$)
- บรรทัดที่ 3 ระบุจำนวนจริงสองจำนวน C และ p ($0 \leq C \leq 10$; $0 < p \leq 1$)
- บรรทัดที่ 4 ระบุจำนวนจริง R , จำนวนเต็ม H และจำนวนจริง S ($1 \leq R \leq 20,000$; $1 \leq H \leq N$; $1 \leq S \leq 20,000$)
- บรรทัดที่ 5 ระบุจำนวนเต็ม T ที่ระบุจำนวนรอบในการจำลองสถานการณ์ ($1 \leq T \leq 100,000$)
- บรรทัดที่ 6 ระบุจำนวนเต็ม $seed$ สำหรับใช้ในการสร้างค่าสุ่ม (จะอธิบายต่อไป)
- อีก N บรรทัดถัดไประบุข้อมูลของประชากร, กล่าวคือ
 - บรรทัดที่ $6 + i$ สำหรับ $1 \leq i \leq N$ ระบุจำนวนจริง X_i และ Y_i และจำนวนเต็ม K_i โดย (X_i, Y_i) เป็นพิกัดเริ่มต้นของของประชากรคนที่ i และ K_i ที่มีค่าเป็น 0 หรือ 1 โดยถ้ามีค่าเป็น 1 หมายความว่าประชากรคนที่ i ติดโรคอยู่
- อีก D บรรทัดถัดไประบุพิกัดของแพทย์แต่ละคน
 - บรรทัดที่ $6 + N + j$ สำหรับ $1 \leq j \leq D$ ระบุจำนวนจริง F_j และ G_j โดย (F_j, G_j)

ข้อมูลส่งออก

โปรแกรมจะต้องเขียนข้อมูลลงในแฟ้ม pandemic.out มีทั้งสิ้น T บรรทัด แต่ละบรรทัด t สำหรับ $1 \leq t \leq T$ ระบุจำนวนคนที่ติดโรคเมื่อจำลองสถานการณ์ไปแล้ว t รอบ

สำหรับการ visualization ถ้าโปรแกรมสร้างแฟ้มข้อมูลระบุตำแหน่งของประชากรและแพทย์ในรูปแบบตามที่ระบุต่อไปนี้ สามารถใช้โปรแกรม viewer ที่ให้มาในการดูภาพของสถานการณ์ได้ โดยแฟ้มดังกล่าวจะมีข้อมูลทั้งสิ้น $T+1$ ชุด (ตั้งแต่เวลา 0 ถึงเวลา T) แต่ละชุดจะประกอบด้วยข้อมูล $N + D$ บรรทัด แต่ละบรรทัดจะระบุพิกัดของคนหรือแพทย์ โดยระบุเป็นจำนวนจริงสองจำนวน แทนพิกัดในแกน x และแกน y และจำนวนเต็มอีกหนึ่งจำนวนโดยมีค่าเป็น 0 (ระบุว่าเป็นประชากรธรรมดา), 1 (ประชากรที่ติดโรค) หรือ 2 (แพทย์) การใช้งาน viewer คู่มือจากรายละเอียดข้อมูลเชิงเทคนิคตอนท้าย

ตัวอย่าง

มีแฟ้มข้อมูลนำเข้าตัวอย่างคือ ex1.in, ex2.in และ ex3.in สำหรับแต่ละแฟ้มจะมีข้อมูลส่งออกตัวอย่างคือ ex1.out, ex2.out, และ ex3.out นอกจากนี้ยังมีแฟ้ม ex1.dat, ex2.dat และ ex3.dat ที่เก็บตำแหน่งของประชากรและแพทย์ในระบบที่สามารถใช้เพื่อดูตัวอย่างการเคลื่อนที่ได้ผ่านทางโปรแกรม viewer

แม้ว่าข้อมูลแฟ้มข้อมูลตัวอย่างจะมีชื่อ ex?.in และ ex?.out ในการทดสอบจริง แฟ้มข้อมูลนำเข้าสำหรับการทดสอบจะชื่อ pandemic.in และโปรแกรมจะต้องสร้างแฟ้ม pandemic.out เสมอ

รายละเอียดของการจำลองสถานการณ์

ในแต่ละรอบ โปรแกรมจะจำลองการเปลี่ยนแปลงของสมาชิกในระบบโดยจะพิจารณาเหตุการณ์หรือกิจกรรมเหล่านี้ตามลำดับ

1. การเคลื่อนที่ของประชากร

ประชากรแต่ละคนจะเคลื่อนที่โดยสุ่มทิศทางในการเคลื่อนที่ (เป็นมุมคิดเทียบจากแกน x ทวนเข็มนาฬิกา) จากนั้นจะคำนวณตำแหน่งใหม่เป็นการเคลื่อนที่ไปในทิศทางดังกล่าวเป็นระยะทาง 1 หน่วย

- ถ้าตำแหน่งใหม่อยู่ในขอบเขตของพื้นที่ (นั่นคือมีพิกัดในแกน x ตั้งแต่ 0 ถึง W และพิกัดในแกน y ตั้งแต่ 0 ถึง H) ประชากรจะย้ายตำแหน่งไปยังพิกัดดังกล่าว
- ถ้าอยู่นอกขอบเขตประชากรคนนั้นจะอยู่ที่ตำแหน่งเดิม

2. การติดต่อของโรค

ถ้าประชากร i ที่ไม่เป็นโรครีระยะห่างจากประชากร f ที่เป็นโรคไม่เกิน C หน่วย มีความน่าจะเป็นที่ i จะติดโรคจาก f เท่ากับ p ถ้า i ติดโรคในการจำลองรอบนี้แล้ว i จะยังไม่สามารถแพร่เชื้อติดไปยังคนอื่นได้ จนกว่าจะถึงรอบถัดไป

ถ้าประชากร i อยู่ในระยะ C หน่วยจากประชากรหลายคนที่เป็นโรค เราจะพิจารณาอย่างเป็นอิสระต่อกันและ i จะติดโรคถ้า i ติดโรคจากประชากรใกล้เคียงคนใดคนหนึ่ง

ยกตัวอย่างเช่น ถ้า ประชากรคนที่ 1 อยู่ในระยะ C หน่วยจากประชากรคนที่ 2 และ 3 ที่ทั้งคู่เป็นโรค สมมติให้ $p = 0.2$ เพื่อความเข้าใจเราจะทดลองคำนวณความน่าจะเป็นที่ 1 จะติดโรคในกรณีนี้ ก่อนอื่นพิจารณาว่าความน่าจะเป็นที่ 1 ไม่ติดโรคคือ 1 ไม่ติดจากทั้ง 2 และ 3 นั่นคือเท่ากับ 0.8×0.8 ดังนั้นความน่าจะเป็นที่จะติดคือ $1 - 0.8 \times 0.8 = 0.36$

อย่างไรก็ตาม ในการเขียนโปรแกรมจริง ๆ ไม่จำเป็นต้องคำนวณความน่าจะเป็นทั้งหมดออกมา เราสามารถสุ่มว่าจะ 1 จะติดจาก 2 หรือไม่ และ 1 จะติดจาก 3 หรือไม่ แยกกัน และนำผลที่ได้มา OR ได้

3. การรักษาของแพทย์

ถ้ามีคน que ที่ติดโรคแล้วที่อยู่ในรัศมี R จากแพทย์คนที่ j ไม่เกิน H คน ทุกคนจะได้รับการรักษาหมดและหายจากการติดโรค ถ้ามีจำนวนคนที่ติดโรคมกกว่า H สมมติว่ามี L คน ความน่าจะเป็นที่แต่ละคนจะได้รับการรักษาจากแพทย์คนที่ j จะเท่ากับ H/L โดยจะพิจารณาคนที่ติดโรคแต่ละคนอย่างเป็นอิสระจากกัน (นั่นคือเป็นไปได้ที่สุดท้ายแพทย์จะรักษา มากกว่าหรือน้อยกว่า H คน)

ถ้าคนที่ติดโรคอยู่ในรัศมี R ของแพทย์หลายคน การที่ผู้ป่วยรายนั้นจะถูกรักษาจากแพทย์แต่ละคนจะพิจารณาอย่างเป็นอิสระต่อกัน และผู้ป่วยจะหายจากโรคถ้ามีแพทย์อย่างน้อยหนึ่งคนรักษา ยกตัวอย่างเช่น ถ้ามีผู้ป่วยที่เป็น ประชากรคนที่ 1 อยู่ในรัศมี R ของแพทย์คนที่ 2 และ 3 ในการนับจำนวนคนที่ติดโรคสำหรับแพทย์ทั้งสองคนจะต้องนับรวมผู้ป่วยคนดังกล่าวด้วย จากนั้นแพทย์แต่ละคนจะพิจารณาว่าจะรักษาประชากรคนดังกล่าวหรือไม่ ถ้าแพทย์คนที่ 2 หรือคนที่ 3 พิจารณาว่าจะรักษา (ไม่ว่าจะเพราะว่ามีผู้ป่วยรวมไม่เกิน H คน หรือว่าผลจากการสุ่มได้ว่าจะรักษาผู้ป่วยคน นี้) ประชากรคนที่ 1 จะหายป่วย เป็นไปได้ที่แพทย์ทั้งสองคนจะเลือกรักษาผู้ป่วยคนดังกล่าวพร้อมกัน

4. การเคลื่อนที่ของแพทย์

ในการเดินทางนั้นแพทย์แต่ละคนจะตั้งเป้าหมายเป็นผู้ป่วยที่จะเดินทางไปรักษา โดยเป้าหมายนี้จะไม่เปลี่ยนจนกว่าผู้ป่วยคนนั้นจะหาย (ซึ่งอาจจะหายเพราะว่าแพทย์คนอื่นรักษาไปก่อนก็ได้) รายละเอียดของการเลือกเป้าหมายและการเดินทางจะเป็นดังนี้

- เริ่มต้นแพทย์ทุกคนจะยังไม่มีเป้าหมาย
- ในแต่ละรอบก่อนการเดินทาง สำหรับแพทย์ที่ไม่มีเป้าหมาย แพทย์จะเลือกเป้าหมายเป็นผู้ป่วยที่อยู่ใกล้ที่สุด ถ้ามีผู้ป่วยอยู่ในระยะที่เท่ากันจะเลือกผู้ป่วยที่มีหมายเลขประชากรน้อยที่สุด (ประชากรคนที่ i มีหมายเลขเท่ากับ i)
- ถ้าไม่มีเป้าหมาย แสดงว่าทุกคนหายป่วยหมดแล้ว แพทย์จะไม่เคลื่อนที่ไปไหน
- ถ้ามีเป้าหมายแล้ว แพทย์จะเคลื่อนที่ไปในทิศทางดังกล่าวให้ใกล้กับเป้าหมายมากที่สุด แต่จะไม่สามารถเคลื่อนที่เป็นระยะทางมากกว่า S หน่วยได้
 - นั่นคือถ้าเป้าหมายอยู่ห่างไม่เกิน S หน่วย แพทย์จะเดินทางไปจุดนั้นพอดี
 - แต่ถ้าเป้าหมายอยู่ห่างเกิน S หน่วย แพทย์จะเดินทางไปในทิศทางของเป้าหมายให้ได้ระยะมากที่สุด นั่นคือจะเคลื่อนที่เป็นระยะทาง S หน่วยพอดี

ข้อมูลเชิงเทคนิค

1. การใช้ไลบรารีสำหรับการสุ่ม

เพื่อความสะดวกและมาตรฐานในการตรวจให้คะแนน โปรแกรมจะต้องใช้ไลบรารีสำหรับการสุ่มที่จัดให้ ซึ่งไลบรารีนี้จะรับรองว่าผลที่ได้จากการสุ่มจะเหมือนกัน ไม่ว่าจะเรียกคำสั่งในการสุ่มในลำดับต่างกันอย่างไร หรือว่าเรียกที่เครื่องใดในกริด

1.1. การประกาศใช้ไลบรารีสำหรับการสุ่ม

ไลบรารีประกอบด้วยแฟ้ม `con_rand.h`, `con_rand.c`, และ `con_rand_full.h` การใช้งานไลบรารีดังกล่าวสามารถทำได้สองวิธีคือ

- ทำโดยการ `include` ส่วนหัว `con_rand.h` ในแฟ้มโปรแกรมหลักจากนั้นลิงก์แฟ้มโปรแกรม `con_rand.c` เข้ากับโปรแกรมหลักตอนคอมไพล์
- สำหรับผู้ที่เขียนโปรแกรมเพียงแฟ้มเดียว และไม่ต้องการลิงก์แฟ้มโปรแกรมในแบบแรก สามารถ `include` ส่วนหัว `con_rand_full.h` ที่มีฟังก์ชันการทำงานทุกอย่างอยู่แล้วได้ อย่างไรก็ตามวิธีนี้ไม่สามารถใช้ได้ถ้าเขียนโปรแกรมแยกเป็นหลาย ๆ ส่วนในหลาย ๆ แฟ้ม เนื่องจากนิยามของฟังก์ชันการสุ่มจะปรากฏหลายที่และจะทำให้ไม่สามารถลิงก์แฟ้มเหล่านั้นเข้าด้วยกันได้

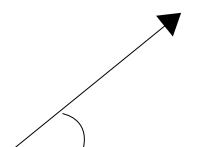
1.2 ข้อจำกัดทั่วไป

- โปรแกรมที่จะเรียกใช้ไลบรารีจะต้องสั่ง `CON_init` หนึ่งครั้งก่อนการใช้งานฟังก์ชันเกี่ยวกับการสุ่มทั้งหมด ถ้าโปรแกรมทำงานที่หลายเครื่อง ฟังก์ชันนี้จะต้องถูกเรียกจากทุก ๆ เครื่องด้วย
- ฟังก์ชันสุ่มทั้งหมดจะรับหมายเลขของรอบการทำงาน โดยรอบการทำงานจะเริ่มนับตั้งแต่ 0 (สังเกตว่าจะเท่ากับเวลาเริ่มต้นของการจำลองสถานการณ์แต่ละรอบ) อย่างไรก็ตามเมื่อเรียกฟังก์ชันสุ่มใด ๆ สำหรับรอบการทำงานที่ t ไปแล้ว จะไม่สามารถเรียกฟังก์ชันสุ่มย้อนกลับรอบการทำงานก่อนหน้า (นั่นคือรอบที่ $t' < t$) ได้อีก
- ฟังก์ชันสุ่มจะรับประกันว่าผลที่ได้จะเหมือนกันสำหรับทุก ๆ โปรแกรม ไม่ว่าลำดับในการเรียกในแต่ละรอบเป็นอย่างไร อย่างไรก็ตาม การย้ายเครื่องที่ทำงานในการเรียกฟังก์ชันสำหรับประชากรหรือแพทย์แต่ละคน เช่น เรียก `CON_rand_movement_angle` สำหรับประชากรคนที่ 1 ที่เครื่องที่ 1 จากนั้นอีก 100 รอบไปเรียกที่เครื่องที่ 2 อาจจะทำให้เสียเวลาเพิ่มขึ้นบ้างเล็กน้อย เนื่องจากไลบรารีจะต้องทำให้การเรียกที่เครื่องที่สองในรอบต่อ ๆ ไปนั้นเหมือนกับการเรียกที่เครื่องที่ 1 อย่างไรก็ตาม เวลาเหล่านี้จัดว่าน้อยมากเมื่อเทียบกับการคำนวณทั้งหมด

1.3 ฟังก์ชันต่าง ๆ

ไลบรารีมีฟังก์ชันดังต่อไปนี้

- `void CON_init(int n, int d, unsigned int seed)`
เป็นฟังก์ชันกำหนดค่าเริ่มต้นให้กับไลบรารี ก่อนทำงานใด ๆ ให้เรียกใช้ฟังก์ชันนี้ก่อน และเรียกเพียงครั้งเดียว โดยอาร์กิวเมนต์ต่าง ๆ มีความหมายดังนี้
 - `n` ระบุจำนวนประชากร
 - `d` ระบุจำนวนแพทย์
 - `seed` คือค่าเมล็ดสำหรับการสุ่ม ค่าดังกล่าวให้นำมาจากแฟ้มข้อมูลป้อนเข้า
- `double CON_rand_movement_angle(int i, int step)`
เป็นฟังก์ชันสำหรับสุ่มทิศทางเคลื่อนที่ของประชากร ฟังก์ชันดังกล่าวจะคืนค่ามุมเป็นเรเดียนโดยคิดเป็นมุมที่ทำกับแกน x ในทิศทางทวนเข็มนาฬิกา ค่าที่คืนจะมีค่าระหว่าง 0



ถึง $2 * M_PI$ ว่า โดยอาร์กิวเมนต์ต่าง ๆ มีความหมายดังนี้

- i ระบุหมายเลขของประชากร (มีค่าตั้งแต่ 1 ถึง N)
 - $step$ ระบุหมายเลขรอบของการจำลองสถานการณ์
- `int CON_rand_is_infected(int i, int f, int step, double pr)`
เป็นฟังก์ชันสำหรับสุ่มว่าประชากรคนที่ i จะติดเชื้อจากประชากรคนที่ f ที่เป็นโรคและอยู่ในระยะ C หรือไม่ โดยอาร์กิวเมนต์ต่าง ๆ มีความหมายดังนี้
 - i ระบุหมายเลขของประชากรที่จะสุ่มว่าจะติดโรคหรือไม่ (มีค่าตั้งแต่ 1 ถึง N)
 - f ระบุหมายเลขของประชากรที่เป็นโรคที่อยู่ในระยะ C จาก i (มีค่าตั้งแต่ 1 ถึง N)
 - $step$ ระบุหมายเลขรอบของการจำลองสถานการณ์
 - pr เป็นความน่าจะเป็นที่จะติด (ตามที่ระบุในข้อมูลป้อนเข้า)
 - `int CON_rand_can_heal(int doc, int i, int step, double pr)`
เป็นฟังก์ชันสำหรับสุ่มว่าแพทย์คนที่ doc จะรักษาประชากรคนที่ f ที่เป็นโรคและอยู่ในระยะ R หรือไม่ โดยอาร์กิวเมนต์ต่าง ๆ มีความหมายดังนี้
 - doc ระบุหมายเลขของแพทย์ (มีค่าตั้งแต่ 1 ถึง D)
 - i ระบุหมายเลขของประชากรที่เป็นโรค (มีค่าตั้งแต่ 1 ถึง N)
 - $step$ ระบุหมายเลขรอบของการจำลองสถานการณ์
 - pr เป็นความน่าจะเป็นที่จะติด (คิดตามที่ระบุในข้อมูลของการจำลองสถานการณ์โดยละเอียดตามด้านบน)

อย่างไรก็ตามในกรณีที่จำนวนผู้ป่วยที่อยู่ในระยะ R จากแพทย์ไม่เกิน H คนไม่จำเป็นต้องใช้ฟังก์ชันนี้ เนื่องจากแพทย์จะสามารถรักษาได้ทุกคนอยู่แล้ว

2. การใช้งาน viewer

ถ้าติดตั้ง python และ wxPython (<http://www.wxpython.org/>) แล้ว เราสามารถเรียก viewer.py ได้โดยสั่ง

```
python viewer.py input-file dat-file
```

โดยที่ `input-file` คือ ชื่อแฟ้มข้อมูลนำเข้า และ `dat-file` ชื่อแฟ้มที่เก็บตำแหน่งของประชากรและแพทย์ในแต่ละรอบ

จากนั้นจะเห็นหน้าต่างแสดงตำแหน่งของสมาชิกในระบบ วงกลมสีฟ้าแสดงประชากรปกติ วงกลมสีแดงแสดงประชากรที่เป็นโรค และวงกลมสีเขียวแสดงแพทย์ ผู้ใช้สามารถกดเมาส์ปุ่มขวาเพื่อให้เริ่มและหยุดการแสดงผล และกดเมาส์ปุ่มซ้ายเพื่อเริ่มแสดงผลใหม่

3. ข้อเสนอแนะอื่น ๆ

ในการพัฒนาและทดสอบโปรแกรม อาจเขียนโปรแกรมให้อ่านข้อมูลนำเข้าและเขียนข้อมูลส่งออกทาง standard input และ output ก่อน เพื่อความสะดวกในการพัฒนา