**Project**

วิชา Data Structure and Algorithms

523231

กลุ่ม G8

หัวข้อ 2C

ผู้จัดทำ

1. นางสาวเอกปวีร์ อุ่นภักดิ์ รหัส B6210236 เลขที่30 เบอร์โทรศัพท์ 0908208456

2. นายเฉลิมเกียรติ คงกะพัน รหัส B6214562 เลขที่ 43 เบอร์โทรศัพท์ 0930101275

3. นางสาวเพชรธนสร พงษ์ศิริยะกุล รหัส B6215415 เลขที่ 47 เบอร์โทรศัพท์ 0823642199

4. นางสาวสุชาวดี เที่ยงตรง รหัส B6238124 เลขที่ 103 เบอร์โทรศัพท์ 0878807717

**User manual**

โปรแกรมหา Shortest path ของกราฟแบบมีน้ำหนักโดยใช้ Dijkstra’s Algorithm

ขั้นตอนการทำงานของโปรแกรม

1. เข้าสู่โปรแกรมและโปรแกรมจะให้ทำการใส่จำนวน node ของกราฟ

2. โปรแกรมจะให้ค่าน้ำหนักจาก node หนึ่งไปยัง node หนึ่ง ซึ่งเราสามารถดูค่าน้ำหนักได้จากกราฟที่เราต้องการหา

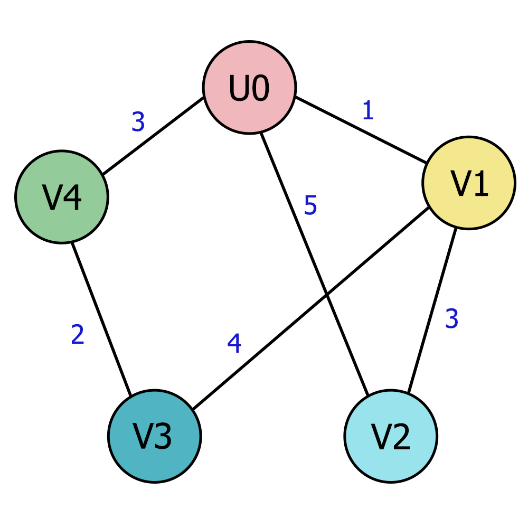
**หมายเหตุ** การป้อนค่าน้ำหนักจะต้องทำการใส่เครื่องหมายลูกน้ำ ( , ) คั่นในแต่ละค่า

2.1) หากทั้งสอง node มี edge เชื่อมถึงกัน ให้เราใส่ค่าน้ำหนักระหว่าง node

2.2) หากทั้งสอง node ไม่มี edge เชื่อมถึงกัน ให้เราใส่ค่า -1

2.3) ป้อนค่าน้ำหนักทั้งหมดลงไปโดยมีเครื่องหมาย , คั่นค่าน้ำหนักไว้ เมื่อครบทุก node แล้ว หน้าจอจะแสดงขั้นตอนวิธีและผลลัพธ์

3. จบการทำงานของโปรแกรม

Ex.

(ภาพประกอบโจทย์)

ตัวอย่างการทำงานของโปรแกรม

Input number of node : 5

Node u0 : 0,1,5,-1,3

Node v1 : 1,0,3,4,-1

Node v2 : 5,3,0,-1,-1

Node v3 : -1,4,-1,0,2

Node v4 : 3,-1,-1,2,0

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

l(u0) v1 v2 v3 v4 Add to S

0 (INFI,-) (INFI,-) (INFI,-) (INFI,-) u0

(1,u0) (5,u0) (INFI,-) (3,u0) v1

(4,v1) (5,v1) (3,u0) v4

(4,v1) (5,v1) v2

(5,v1) v3

Path : u0->v1 , u0->v4 , v1->v2 , v1->v3

**ส่วนของการอธิบาย Code**

(รูปภาพประกอบสามารถดูได้ด้านล่างหรือในไฟล์ชื่อ File code)

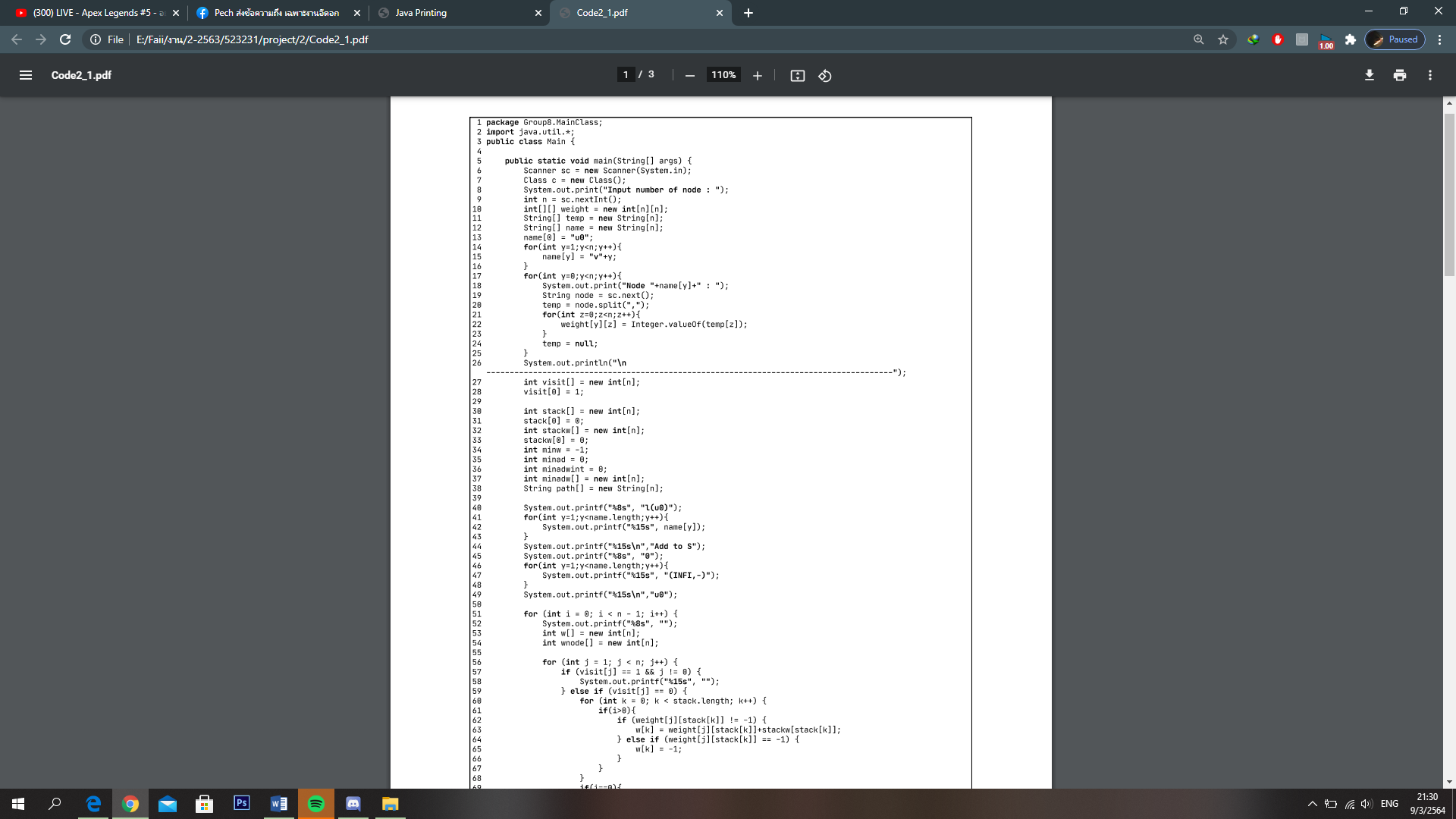
Class Main

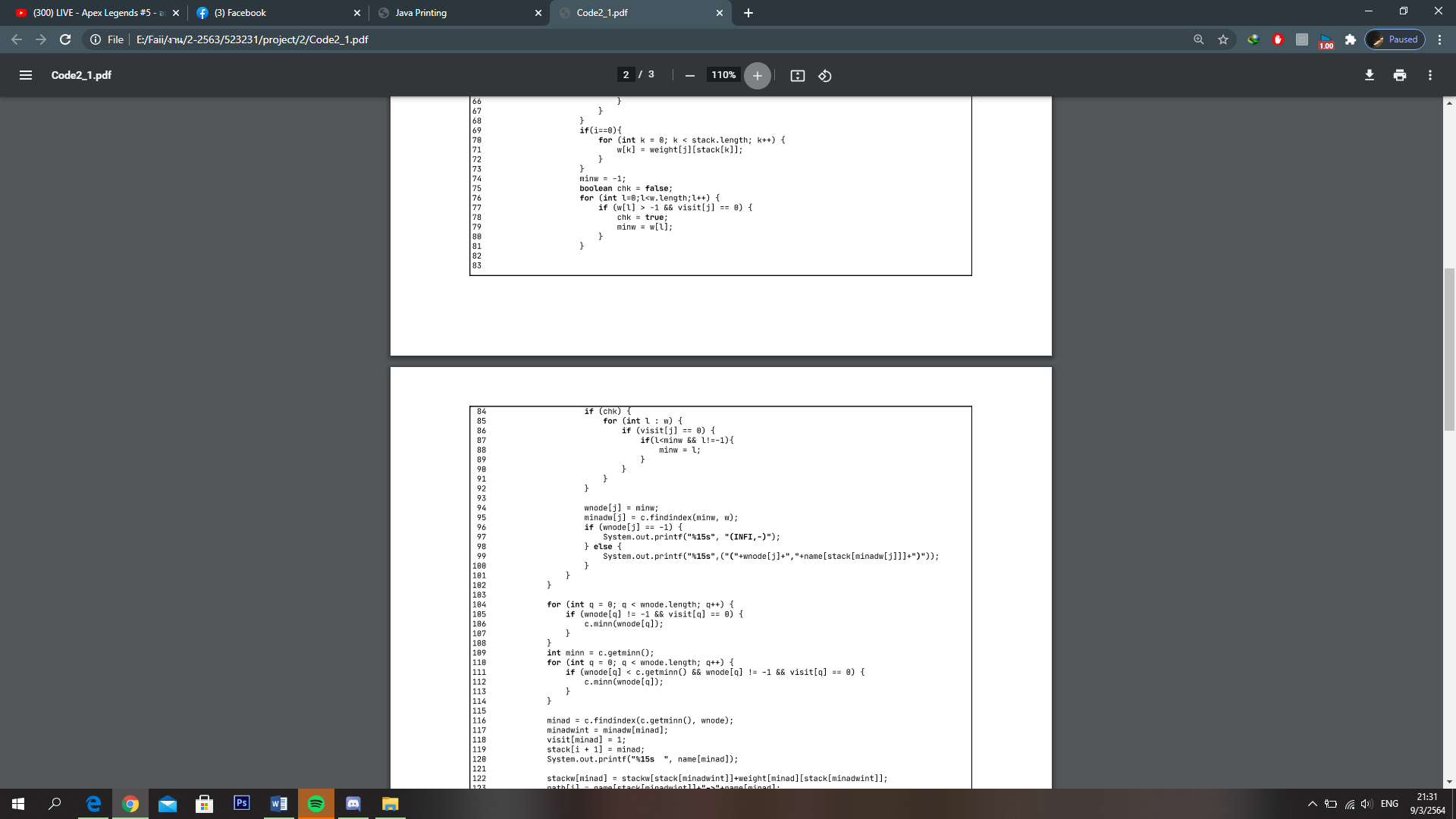
|  |  |
| --- | --- |
| บรรทัด | คำอธิบาย |
| 6-11 | ประกาศตัวแปรชื่อ sc รับค่าจากแป้นพิมพ์ สร้าง object จากคลาส Class ประกาศตัวแปร  ประเภท integer ชื่อ n ตัวแปร array2 มิติชื่อ weight ของข้อมูลประเภทinteger ประกาศ  ตัวแปรarrayชื่อtempของข้อมูลประเภทstring |
| 12-13 | ประกาศตัวแปร array ชื่อว่า name ของข้อมูลประเภท string มีขนาดเท่ากับจำนวน node ที่  รับเข้ามาเพื่อเก็บค่าชื่อของ node โดย node เริ่มต้นนั้นจะมีชื่อว่า u0 เสมอ |
| 14-16 | วน loop for เพื่อกำหนดชื่อของ node ที่เหลือ ถ้าหากมี 5 node จะได้ชื่อ 4 node ดังนี้  v1 คือชื่อ node ที่ 2 , v2 คือชื่อ node ที่ 3 , v3 คือชื่อ node ที่ 4 , v4 คือชื่อ node ที่ 5 |
| 17-25 | รับค่าน้ำหนักของแต่ละ node ดังตัวอย่างด้านบนคือ  node u0 ไปที่ u0 (ตัวเอง) จะให้ค่าเป็น 0  u0 ไป v1 มีค่าน้ำหนัก = 1  u0 ไป v2 มีค่าน้ำหนัก = 5  u0 ไม่มี edgeเ ชื่อมไปที่ v3 ดังนั้นจะให้ค่าน้ำหนัก = -1  u0 ไป v4 มีค่าน้ำหนัก = 3  จากข้อมูลด้านบนทำให้เราจะใส่ค่าของ node0 ว่า 0,1,5,-1,3 โดยใช้ .split(“,”) ในการตัดแบ่งข้อความแล้วจัดเก็บค่าที่ได้เป็น array จากนั้นนำค่าที่ได้วนลูปเพื่อแทนค่านั้นลงไปใน array2 มิติชื่อว่า weight โดยค่าที่ได้นั้นจะแทนไปในแต่ละ column เมื่อครบทุกตัวแล้วจะให้ temp=null เพื่อทำให้ array ของตัวแปร temp กลับมาว่างอีกครั้งเพื่อรับค่าใหม่ จากนั้นจึงกลับไปที่  for loop แรกเพื่อรับค่าน้ำหนักของ node ต่อไป |
| 26 | print output ออกเพื่อคั่นส่วนของวิธีทำและผลลัพธ์ออกจากขั้นตอนการรับค่า |
| 27-28 | ประกาศตัวแปรประกาศตัวแปร array ชื่อ visit ของข้อมูลประเภท integer มีขนาดเท่ากับจำนวนของ node ที่รับเข้ามา ซึ่งตัวแปรนี้ใช้ในการเช็คว่า node ใดที่ถูกจัดเก็บเข้า stack แล้ว ถ้ามีค่าเท่ากับ 1 คืออยู่ใน stack แล้ว ถ้ามีค่าเท่ากับ 0 คือยังไม่อยู่ใน stack และกำหนดให้ visit[0] = 1 เพราะ node u0 เป็น node เริ่มต้น จึงจัดเก็บเข้า stack แล้วเสมอ |
| 30-31 | ประกาศตัวแปรarrayชื่อว่า stack ของข้อมูลประเภท integer มีขนาดเท่ากับจำนวนnodeที่รับเข้ามา โดยให้ stack[0] = 0 |
| บรรทัด | คำอธิบาย |
| 32-33 | ประกาศตัวแปรarrayชื่อว่า stackw ของข้อมูลประเภทinteger มีขนาดเท่ากับจำนวนnodeที่รับเข้ามา ทำหน้าที่เก็บสะสมค่าน้ำหนักของnodeที่ต้องการหาไปจนถึงnodeเริ่มต้น และกำหนดให้ stackw[0] = 0 คือnodeเริ่มต้นนั้นมีน้ำหนักเท่ากับ0 |
| 34-36 | ประกาศตัวแปรประเภท integer ชื่อว่า minw , minad และ minadwint โดย  กำหนดให้ minw มีค่าเท่ากับ -1 , minad มีค่าเท่ากับ 0 และ minadwint มีค่าเท่ากับ 0 |
| 37 | ประกาศตัวแปร array ชื่อว่า minadw ของข้อมูลประเภท integer มีขนาดเท่ากับจำนวน  node ที่รับเข้ามา ไว้เก็บ node ที่ node(j) แต่ละตัวเชื่อมอยู่ |
| 38 | ประกาศตัวแปร array ชื่อว่า path ของข้อมูลประเภท string มีขนาดเท่ากับจำนวน node ที่  รับเข้ามา โดยทำหน้าที่เก็บค่าว่า node ที่เราต้องการนั้นมีเส้นทางผ่านมาจาก node ใด |
| 40-44 | Print บรรทัดแรกของส่วนวิธีทำ ใช้ for loop เพื่อ print ชื่อ node ตัวที่สองขึ้นไป |
| 45-49 | Print บรรทัดที่สองของส่วนวิธีทำ ใช้ for loop สำหรับ node ที่สองขึ้นไปเพื่อ print ค่า  น้ำหนักและเส้นทางที่ได้คือ (INFI,-) เพราะในขั้นตอนนี้จะพบแค่ node เริ่มต้นคือ u0 ทำให้  node อื่น ๆ จะมีค่าน้ำหนักเป็น infinity และยังไม่มีเส้นทาง |
| 51-125 | ใช้ for loop(int i) เพื่อเข้าโปรแกรมและแสดงผลวิธีทำจนทุก node ถูกเก็บเข้าไว้ในสแตก |
| 52 | Print ช่องว่างที่ตรงกับ u0 เพราะ u0 ถูกเก็บเข้าสแตกแล้ว จึงไม่ต้องทำต่อ |
| 53 | ประกาศตัวแปร array ชื่อว่า w ของข้อมูลประเภท integer มีขนาดเท่ากับจำนว นnode ที่รับ  ค่าเข้ามา ไว้เก็บค่าที่ node(j) ต่อกับทุกตัวในสแตก |
| 54 | ประกาศตัวแปร array ชื่อว่า wnode ของข้อมูลประเภท integer มีขนาดเท่ากับจำนวน  Node ที่รับค่าเข้ามา ไว้เก็บค่าน้ำหนักที่น้อยที่สุดของแต่ละ node |
| 56-59 | ใช้ for loop (int j) เพื่อเข้าถึงตำแหน่งของ node หาก node นั้นถูกจัดเก็บเข้าสแตกแล้วจะ  เข้าเงื่อนไข if ซึ่งจะ print ช่องว่างออกมาเพราะ node นั้นอยู่ในสแตกแล้วไม่ต้องทำต่อ  หากยังไม่อยู่ในแสตกจะเข้าเงื่อนไข else if |
| 59-61 | เมื่อพบว่า node นั้นยังไม่อยู่ในสแตกจะเข้า for loop (int k) เพื่อหาค่าในแต่ละเส้นทางของ  Node นั้น ถ้า i>0 จะเข้าไปทำในเงื่อนไข if |
| 62-63 | ถ้ามี edge เชื่อมระหว่าง node(j) กับ node(k) จะเข้าไปทำในเงื่อนไข if คือค่าของ w ณ  ตำแหน่งที่ k จะมีค่าเท่ากับ ค่าน้ำหนักของ edgeที่ เชื่อมระหว่าง node(j) กับ node(k) + ค่า  น้ำหนักสะสมของ node(k) ที่เชื่อมไปถึง node เริ่มต้น |
| 64-65 | ถ้าไม่มี edge เชื่อมระหว่าง node(j) กับ node(k) จะเข้าไปทำในเงื่อนไข else if คือค่าของ w  ณ ตำแหน่งที่ k จะมีค่าเท่ากับ -1 คือ infinity |
| 69 | ถ้า node นั้นยังไม่อยู่ในสแตกและ i=0 จะเข้าไปทำในเงื่อนไข if |
| 70-72 | ใช้ for loop (int k) เพื่อเก็บค่าน้ำหนักของ node(k) ที่ต่อกับ node เริ่มต้น (u0) ใน w[k] |
| บรรทัด | คำอธิบาย |
| 74 | ให้ตัวแปร integer minw = -1 เพื่อรีเซ็ตค่า |
| 75 | ประกาศตัวแปร boolean ชื่อว่า chk ให้มีค่าเป็น false |
| 76 | ใช้ for loop (int l) เพื่อหาค่า minw ที่ไม่เท่ากับ -1 เพื่อใช้ในการเปรียบเทียบต่อไป |
| 77-81 | ถ้าค่า w ณ ตำแหน่งที่ l มีค่ามากกว่า -1 และ node(j) ยังไม่ถูกเก็บเข้าสแตก จะเข้าไปทำใน  เงื่อนไข if คือให้ chk = true และให้ minw มีค่าเท่ากับค่าจาก array w ตัวสุดท้ายที่ไม่ใช่ -1 |
| 84 | ถ้า chk = true นั่นคือใน array w มีค่าที่ไม่เท่ากับ -1 อยู่ |
| 85 | ใช้ for loop (int l) เพื่อนำค่าจาก array w ออกมา |
| 86 | ถ้า node(j) ยังไม่ถูกเก็บเข้าสแตก จะเข้าไปทำในเงื่อนไข if |
| 87 | ถ้าค่า l น้อยกว่า minw และ l ไม่เท่ากับ -1 จะเข้าไปทำในเงื่อนไข if คือให้ minw มีค่าเท่ากับ  l นั้น |
| 94 | ให้ค่าใน array wnode ณ ตำแหน่งที่ j มีค่าเท่ากับ minw นั่นคือให้ wnode[j] เก็บค่าน้อย  ที่สุด |
| 95 | หาตำแหน่งของ node ที่ node(j) เชื่อมต่ออยู่ โดยการหาตำแหน่งของ minw ใน w[] นั้น  จะต้องส่งค่า minw และ w เข้าไปใน c.findindex(minw , w); |
| 96-97 | ถ้า wnode ณ ที่ j มีค่าเท่ากับ -1 คือ ถ้า wnode ที่ node(j) ไม่เชื่อมต่อกับตัวใดในสแตกเลย  จะ output ว่า (INFI, -) |
| 98-99 | ถ้า wnode ที่ node(j) มีการเชื่อมต่อกับกับตัวในสแตก จะ output ว่า (ค่าน้ำหนักที่น้อยที่สุด  ของ node(j) ไปจนถึง node เริ่มต้น , ชื่อของ node ที่เชื่อมอยู่กับ node(j) ที่ทำให้ได้ค่า  น้ำหนักที่น้อยที่สุด) |
| 104 | ใช้ for loop (int q) เพื่อหาค่า minn ที่ไม่เท่ากับ -1 เพื่อนำมาใช้เปรียบเทียบ |
| 105-107 | ถ้าค่าของ wnode[q] ไม่เท่ากับ -1 และ node(q) ยังไม่อยู่ในสแตก ให้ส่งค่า wnode[q] เข้า  ไปเก็บไว้ที่ c.minn(wnode[q]) นั่นคือ c.minn(); จะเก็บค่าจาก array wnode ตัวสุดท้ายที่  ไม่เป็น -1 |
| 109 | กำหนดให้ integer minn รับค่าจาก c.getminn(); |
| 110 | ใช้ for loop (int q) เพื่อหาค่าน้อยที่สุดในรอบนั้น ๆ |
| 111-113 | ถ้าค่าของ wnode[q] น้อยกว่าค่าของ c.getminn(); และ ค่าของ wnode[q] ไม่เท่ากับ -1  และ node(q) ยังไม่อยู่ในสแตก ให้ส่งค่า wnode[q] ไปเก็บไว้ใน c.minn(); |
| 116 | หาตำแหน่งของค่า c.getminn(); ใน wnode[] โดยการส่งค่าเข้าไปใน  c.findindex(c.getmin() , wnode); และ minad จะเท่ากับค่าที่หาได้ |
| บรรทัด | คำอธิบาย |
| 117 | ให้ minadwint เก็บค่าตำแหน่งของ node ที่ node(minad) เชื่อมอยู่ |
| 118 | จัดเก็บ node(minad) เข้าสแตก |
| 119 | ให้ stack[i+1] มีค่าเท่ากับ minad |
| 120 | output ชื่อของ node ที่ถูกจัดเก็บเข้าสแตก |
| 122 | ค่าน้ำหนักของ node ที่ถูกจัดเก็บเข้าสแตกจะเท่ากับ (ค่าน้ำหนักสะสมของ node ที่เชื่อมอยู่  กับ node(minad) + ค่าน้ำหนักของnode(minad) ไป node ที่เชื่อมกันอยู่) |
| 123 | ให้ path[i] เก็บชื่อเส้นทางของnodeที่ถูกนำเข้าสแตกแล้ว |
| 124 | output ให้ขั้นบรรทัดต่อไปเพื่อแสดงผลวิธีทำขั้นต่อไป |
| 127-132 | แสดงผลลัพธ์เส้นทางของแต่ละ node และจบการทำงานของโปรแกรม |

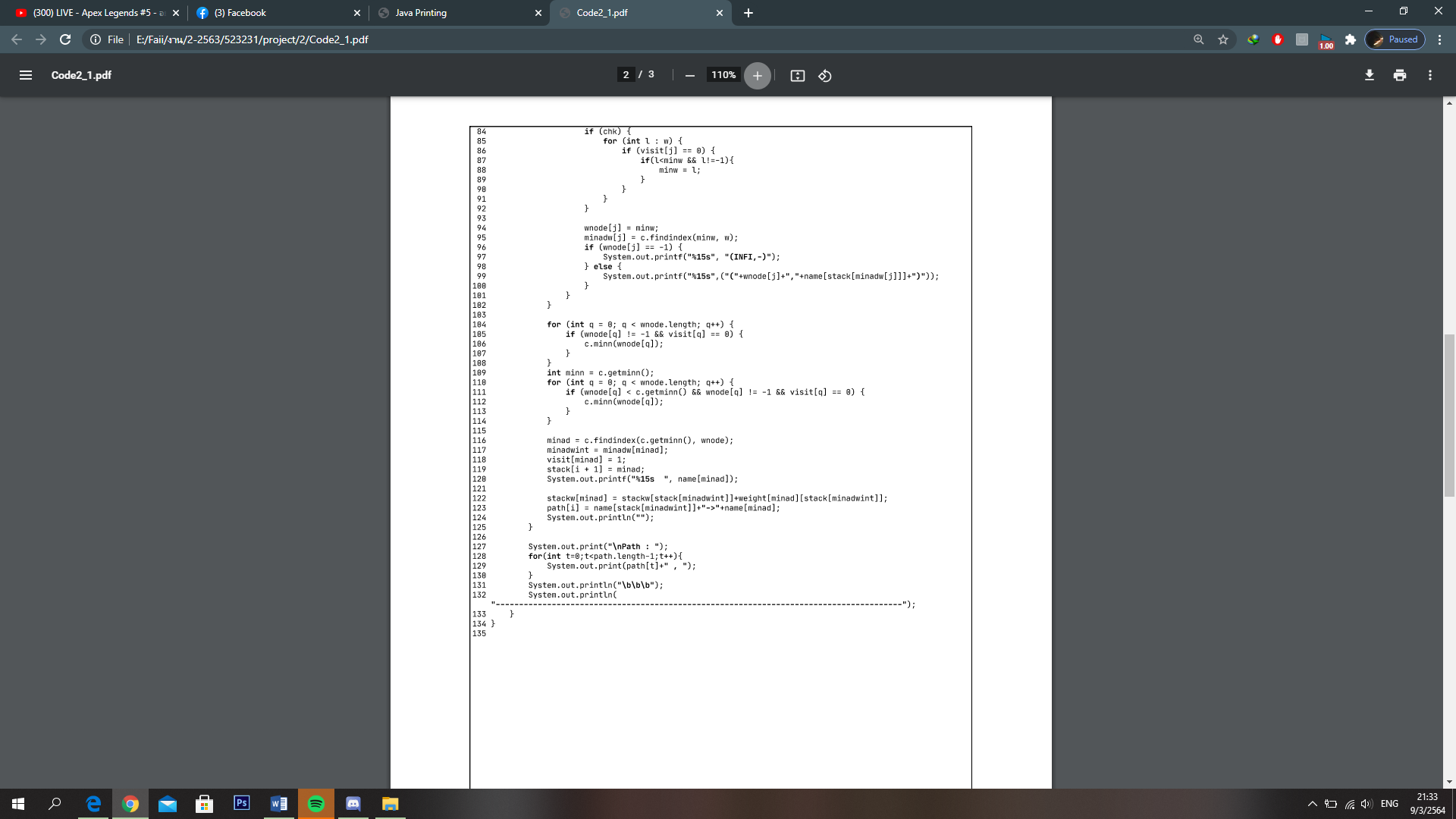
Class Class

|  |  |
| --- | --- |
| บรรทัด | คำอธิบาย |
| 4-6 | ประกาศตัวแปรแบบ private ชื่อว่า minn , minad , min เป็นตัวแปรประเภท integer |
| 8-11 | method min รับค่า integer เข้ามาแล้วเซตค่า |
| 13-16 | method getmin ใช้ส่งค่า min ที่เซตแล้วกลับไป |
| 18-21 | method minn รับค่า integer เข้ามาแล้วเซตค่า |
| 23-26 | method getminn ใช้ส่งค่า minn ที่เซตแล้วกลับไป |
| 28-31 | method minad รับค่า integer เข้ามาแล้วเซตค่า |
| 33-36 | method getminn ใช้ส่งค่า minad ที่เซตแล้วกลับไป |
| 38-50 | method findindex(int m, int[] n); ใช้หาตำแหน่ง |
| 39 | ประกาศตัวแปรประเภท integer ชื่อ index โดยกำหนดให้มีค่าเท่ากับ 0 |
| 40 | ประกาศตัวแปรประเภท integer ชื่อ i โดยกำหนดให้มีค่าเท่ากับ 0 |
| 41 | ประกาศตัวแปรประเภท Boolean ชื่อ chk โดยกำหนดให้ chk มีค่าเป็น false |

Class Main







Class Class

