บทที่ 2

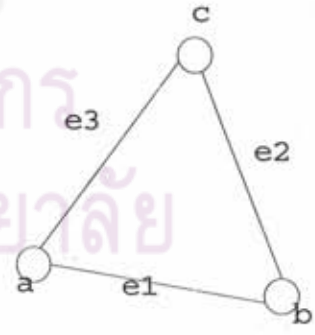
ทฤษฎีและแนวคิดที่นำมาใช้ในการวิจัย

ในบทนี้จะกล่าวถึงทฤษฎีกราฟเบื้องต้น และ อัลกอริทึมที่ใช้ในการวิจัย เพื่อให้ ผู้ที่สนใจได้ทำความเข้าใจถึงคำจำกัดความ และ ขั้นตอนการทำงานของอัลกอริทึม ซึ่งเมื่อไปใช้โปรแกรมจะทำให้เกิดความเข้าใจได้มาก และรวดเร็วยิ่งขึ้น

ทฤษฎีกราฟ

กราฟ[1] (Graph) คือ คู่ลำดับของเซต (V, E) โดยที่ V คือเซตของจุด(vertex) ต่าง ๆ ของกราฟ และ E คือเซตของเส้นเชื่อม (edge) ระหว่างคู่ของจุดในกราฟ ตัวอย่างเช่น กราฟ G = (V, E) ในรูปที่ 2.1 มี V = {a, b, c} และ E = { {a, b}, {a, c}, {b, c} }

รูปที่ 2.1 ตัวอย่างกราฟ



รูปที่ 2.2 ตัวอย่างกราฟที่กำหนดเส้นเชื่อม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

---

[1] สมชาย ประสิทธิ์จูตระกูล,วิทยา วัชระวิทยากุล. คณิตศาสตร์ดีสครีต.เชิงประยุกต์. บริษัทซีเอ็ดยูเคชั่น จำกัด, 2521. หน้า 160.

นอกจากนี้เราสามารถใช้คู่ลำดับของจุดแทนเส้นเชื่อม เช่น { a,b } แทนด้วย e1 ดังในรูปที่ 2.2

เราเรียก จุดสองจุดว่าต่อกัน (adjacent) ถ้ามีเส้นเชื่อมเส้นหนึ่งเชื่อมโยงระหว่างจุดทั้งสอง และ เรียกจุดที่เป็นจุดปลายของเส้นเชื่อมหนึ่งว่าติดกับ (incident) เส้นเชื่อมนั้น เช่นในรูปที่ 2.2 a ต่อกับ b ในขณะที่ a ติดกับ e1 และ e3 และเราเรียกเส้นเชื่อม 2 เส้นว่าอยู่ติดกัน การลำดับของจุดและเส้นเชื่อมที่ต่อกันเป็นทางโดยผ่านเส้นเชื่อมที่ต่างกันและจุดไม่ซ้ำกันเราเรียก ลำดับที่สลับกันระหว่างจุดกับเส้นเชื่อมที่ติดต่อกันว่า ทางเดิน (path) และทางเดินที่จุดแรกและจุดสุดท้ายเป็นจุดเดียวกัน เราเรียกว่า วงจร (circuit) เช่นในรูปที่ 2.2 a-e1-b-e2-c คือทางเดินทางหนึ่ง และ a-e1-b-e2-c-e3-a คือวงจรหนึ่งของกราฟ

กราฟอัลกอริทึม

อัลกอริทึมทางกราฟมีมากมายที่ใช้ในการแก้ไขปัญหาทาง กราฟเช่น การเดินทางไปยังทุกๆ เมืองโดยใช้ระยะทางสั้นที่สุดเป็นต้น ซึ่งจะเห็นถึงการนำกราฟมาประยุกต์ใช้งานกับชีวิตจริงได้ สำหรับในโครงงานวิทยานิพนธ์นี้จะกล่าวถึง เฉพาะส่วนที่จะออกแบบ และ พัฒนาระบบจินตทัศน์อัลกอริทึมทางกราฟเท่านั้น ซึ่งมีดังต่อไปนี้

2.1 อัลกอริทึมการแวะผ่าน(Traversal Algorithm)

จะพิจารณา 2 อัลกอริทึมที่ใช้สำหรับการไปเยี่ยมจุดทุกๆ จุดโดยที่ลักษณะการเยี่ยมจุดแต่ละจุดจะมีรูปแบบที่แตกต่างกันมีดังนี้

2.1.1 การค้นหาในแนวลึก (Depth-First Search)

ลักษณะการค้นหาจะเริ่มจากจุดใดจุดหนึ่งในกราฟมาเป็นราก (root) จากนั้นเริ่มสร้างทางเดินตามเส้นเชื่อมในกราฟจากราก ไปยังจุดอื่นที่อยู่ในระดับ(level) ต่ำกว่า ไปจนกว่าจะไม่สามารถหาเส้นเชื่อมที่ต่อเป็นทางเดินได้ โดยไม่ใช้จุดที่อยู่บนทางเดินใด ๆ ที่หาได้ก่อนหน้านี้จุดสุดท้ายของทางเดินที่หาได้จะเป็นใบ (leaf) ของต้นไม้ (tree) จากนั้นก็กลับขึ้นไปที่จุดระดับพ่อ/แม่ (Parent) ของใบนั้น แล้วเริ่มค้นหาจุดอื่นต่อไป ทำเช่นนี้เรื่อยไปจนกว่าจะไม่มีเส้นเชื่อมให้เลือก

ในการค้นหาจุดและเยี่ยมจุดทุก ๆ จุดนั้นเราใช้โครงสร้างข้อมูลแบบกองซ้อน (stack) มาช่วยในการทำงาน ดังมีขั้นตอนต่อไปนี้

```
Void  DepthFirstSearch( int k )
      {
          val[k] = TRUE ;
          push( all w adjacent list to k )
          while( Stack not empty )
          {
             T = pop() ;
             val[T] = TRUE ;
             Visit(T) ;
             for( all w adjacent list to T )
                if( !val[w] ) ;
                {
                    push(w) ;
                    val[w] = TRUE ;
                }
          }
      }
```

2.1.2 การค้นหาในแนวกว้าง (Breadth-First Search)

เริ่มด้วยการเลือกจุดใดจุดหนึ่งในกราฟมาเป็นราก จากนั้นเริ่มแวะจุดที่อยู่ในระดับที่หนึ่งจนครบ จากนั้นก็จะแวะจุดที่อยู่ในระดับที่สองจนครบ ทำเช่นนี้เรื่อยไปจนครบทุกจุด

หลักการทำงานได้นำโครงสร้างแบบแถวคอย(queue) มาช่วยมีขั้นตอนดังนี้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

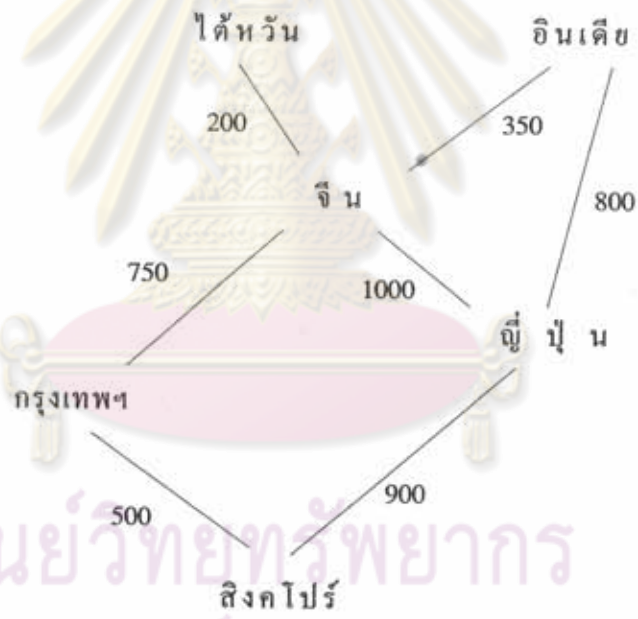
```
Void BreadthFirst( Graph_type G )
   {
       Vertexqueue_type  Q ;
       Boolean_type      visited[MAX] ;
       int                   v,w ;

       for( all v in G )
          visited[v] = FALSE ;
       initialize(Q)  ;          /* Set the queue
                                    to be empty */
       for( all v in G )
        if( !visited[v] )
          {
             AddQueue(v,Q) ;
             do {
                 DeleteQueue(v,Q) ;
                 visited[v] = TRUE ;
                 Visit(v) ;
                 for( all w adjacent to v )
                   if( !visited[w] )
                      AddQueue(w) ;
               } while ( !Empty(Q) ) ;
          }
   }
```

2.2 กราฟมีน้ำหนัก(Weighted Graph) หรือข่ายงาน(Network)

ลักษณะกราฟโดยทั่วไปจะประกอบไปด้วยจุดและเส้นอยู่แล้ว แต่เมื่อมีการกำหนดค่าตัวเลขที่เรียกว่า น้ำหนัก(weight) กำหนดไว้ตามเส้นเชื่อมต่างๆบนกราฟ เราเรียกกราฟดังกล่าวว่ากราฟมีน้ำหนักซึ่งค่าตัวเลขอาจจะมีหน่วยวัดต่างๆ เช่น เมตร,นิ้ว เป็นต้น จากกราฟมีน้ำหนักนี้เองทำให้สามารถนำมาประยุกต์ใช้กับชีวิตจริงได้ เช่น เส้นทางถนนในกรุงเทพฯ ถูกจำลองได้โดยใช้กราฟมีน้ำหนัก ที่มีจุดของกราฟแทนทางแยก และมีระยะทางเป็นน้ำหนักของเส้นเชื่อม จากนั้นเราจึงตั้งคำถามเช่น การหาทางเดินสั้นที่สุดระหว่างสถานที่ต่างๆในกรุงเทพฯ โดยหาจากกราฟที่สร้างขึ้น ดังนั้นวิธีการในการแก้ปัญหาที่เกี่ยวข้องกับกราฟมีน้ำหนักมีดังนี้

2.2.1 ทางเดินสั้นที่สุด[2] (Shortest Path)

การหาทางเดินที่สั้นที่สุดระหว่างคู่จุดใดๆในข่ายงาน เป็นปัญหาซึ่งเกิดขึ้นบ่อยมากโดยในที่นี้ความยาวของทางเดินถูกกำหนดให้เป็นผลรวมของน้ำหนักของเส้นเชื่อมซึ่งประกอบเป็นทางเดินนั้น เช่นเส้นทางการบินของสายการบินหนึ่งถูกจำลองได้โดยใช้ข่ายงานซึ่งมีจุดแทนเมือง เส้นเชื่อมระหว่างเมืองแทนเส้นทางบิน และมีน้ำหนักของเส้นเชื่อมซึ่งอาจแทนระยะทาง เวลาในการเดินทาง หรือ ราคาค่าโดยสารของเส้นทางบิน การหาทางเดินซึ่งมีผลรวมของน้ำหนักน้อยที่สุด (เรียกว่าทางเดินสั้นที่สุด) ระหว่างเมืองสองเมือง จะบอกถึงคุณสมบัติบางประการของทางเดินที่หาได้ ซึ่งขึ้นกับความหมายของน้ำหนักของเส้นเชื่อมอาจเป็นระยะทางสั้นที่สุด เวลาการเดินทางเร็วที่สุด หรือราคาค่าโดยสารรวมถูกที่สุด เป็นต้น ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 ข่ายงานแสดงราคาค่าโดยสารของสายการบินหนึ่ง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

อัลกอริทึมในการหาทางเดินสั้นที่สุดบนข่ายงานในโครงงานนี้ ได้ถูกคิดค้นโดยนักคณิตศาสตร์ชาวดัทช์ชื่อดิสตรา(Dijkstra) เมื่อปี ค.ศ 1959 ในการหาทางเดินสั้นที่สุดของดิสตรานั้นได้กำหนดให้

[2] สมชาย ประสิทธิ์ยุควระกุล,วิทยา วัชระวิทยากุล. คณิตศาสตร์ดีสครีต.เชิงประยุกต์. หน้า 255-258.

1. d_{ij} คือ ความยาวของเส้นเชื่อมระหว่างจุด i และ j ในกรณีที่จุด i และ จุด j ไม่ต่อกันให้ $d_{ij} = \infty$
2. S เป็นเซตซึ่งเก็บจุดประเภทที่ทราบทางเดินจากจุดเริ่มต้นสั้นที่สุด
3. ณ. จุด v ใดๆจะมีคู่ลำดับ (P_v, L_v) กำกับอยู่โดยที่

   L_v ของจุดประเภทที่ทราบทางเดินจากจุดเริ่มต้นสั้นที่สุดแล้ว เป็นระยะทางสั้นที่สุดจากจุดเริ่มต้นถึงจุด v

   L_v ของจุดประเภทที่ถูกแวะผ่านแล้ว บอกถึงระยะทางจากจุดเริ่มต้นถึงจุด v ที่สั้นที่สุดที่ได้พบมา (ให้สังเกตว่าอาจมีทางเดินอื่นจากจุดเริ่มต้นถึงจุด v ที่สั้นกว่า L_v ซึ่งยังหาไม่พบ)

   L_v ของจุดประเภทที่ยังไม่ถูกแวะผ่านจะเป็น ∞ เนื่องจากว่าเป็นจุดซึ่งยังไปไม่ถึง

   P_v คือ จุดก่อนหน้าจุด v ซึ่งอยู่บนทางเดินจากจุดเริ่มต้นถึงจุด v สั้นที่สุดที่ได้พบมา

ขั้นตอนการทำงานในการหาทางเดินสั้นที่สุดระหว่างจุด a และ z ในข่ายงานด้วยวิธีของดิสตรามีดังนี้

1. ให้ $S = \{a\}$,
2. สำหรับแต่ละจุด v ที่ไม่อยู่ใน S

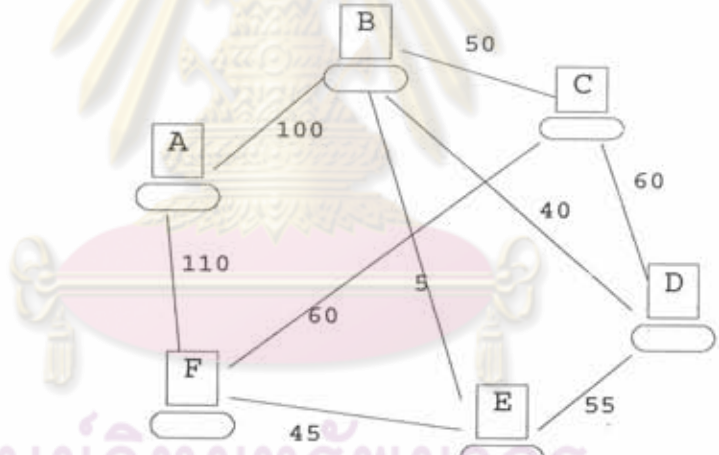
   ถ้า $L_v > L_i + d_{iv}$ ให้ $L_v = L_i + d_{iv}$ และ ให้ $P_v = i$

   โดยที่จุด i คือ จุดซึ่งเป็นสมาชิกของเซต S จุดล่าสุด
3. ให้จุด v ที่ไม่อยู่ใน S และที่มีค่า Lv น้อยที่สุดเป็นสมาชิกใหม่ของ S
4. ถ้า v คือจุด z ที่เป็นจุดเป้าหมายก็จะได้ Lz เป็นระยะทางสั้นที่สุดจากจุด a ไป z ถ้า v ไม่ใช่ z ให้กลับไปทำขั้นตอนที่ 2 และ 3 ซึ่งจนกระทั่ง จุดที่เป็นสมาชิกล่าสุดของ S คือจุดเป้าหมาย z

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2.2.2 ต้นไม้แบบทอดข้ามที่เล็กที่สุด[3] (Minimum Spanning Tree)

สมมติให้บริษัทแห่งหนึ่งมีความประสงค์จะต่อศูนย์คอมพิวเตอร์ ซึ่งกระจายตามเมืองต่าง ๆ เข้าด้วยกัน ด้วยการเช่าสายโทรศัพท์เพื่อรับส่งข้อมูล ในกรณีที่ทางองค์การโทรศัพท์มีคู่สายให้เช่าระหว่างเมืองในราคาที่ต่างกันดังแสดงในรูปที่ 2.4ก อยากทราบว่าทางบริษัทควรจะเช่าคู่สายใดบ้างเพื่อให้ศูนย์คอมพิวเตอร์ทุกๆศูนย์ติดต่อถึงกันได้ ด้วยค่าใช้จ่ายของสายโทรศัพท์ที่ต่ำที่สุด ปัญหานี้สามารถถูกจำลองด้วยข่ายงาน ซึ่งมีจุดแทนศูนย์คอมพิวเตอร์ เส้นเชื่อมแทนสายโทรศัพท์ระหว่างศูนย์ซึ่งองค์การโทรศัพท์สามารถต่อให้ได้ และมีราคาค่าเช่าสายโทรศัพท์เป็นน้ำหนักกำกับบนเส้นเชื่อมแต่ละเส้น เนื่องจากต้นไม้แบบทอดข้ามของกราฟใดๆจะประกอบไปด้วยจุดทุก ๆ จุดบนกราฟนั้น และจากคุณสมบัติของต้นไม้จะมีทางเดินระหว่างทุกๆคู่จุดในต้นไม้ ดังนั้นคำตอบของปัญหานี้คือ การหาต้นไม้แบบทอดข้ามของข่ายงานนี้ ซึ่งมีผลรวมของน้ำหนักของเส้นเชื่อมของต้นไม้แบบทอดข้ามที่น้อยที่สุด เราเรียกต้นไม้ดังกล่าวว่าต้นไม้แบบทอดข้ามที่เล็กที่สุด ดังแสดงในรูปที่ 2.4ข

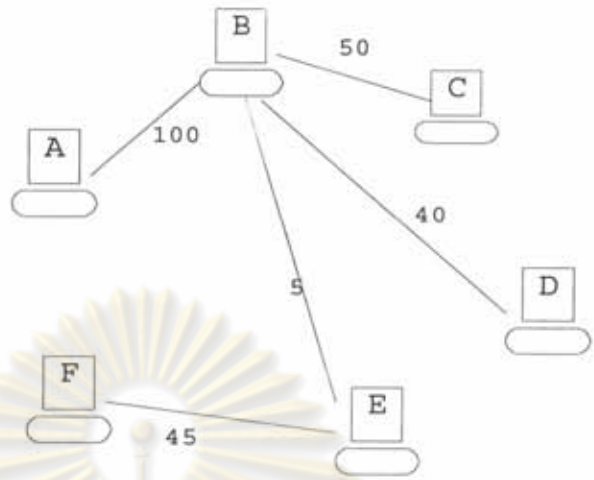


(ก)

รูปที่ 2.4 ข่ายงานเครื่องคอมพิวเตอร์และต้นไม้ทอดข้ามที่เล็กที่สุด

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

[3] สมชาย ประสิทธิ์จูตระกูล,วิทยา วัชระวิทยากุล. คณิตศาสตร์ดีสครีต เชิงประยุกต์. หน้า 263-265.



(ข)

รูปที่ 2.4 ข่ายงานเครื่องคอมพิวเตอร์และต้นไม้ทอดข้ามที่เล็กที่สุด (ต่อ)

อัลกอริทึมในการหาต้นไม้แบบทอดข้ามที่เล็กที่สุด ที่นิยมใช้กันมากมีอยู่ 2 วิธี คือ

1. อัลกอริทึมของครูสกัล

โจเซฟ เบอร์นาร์ด ครูสกัล(Joseph Bernard Kruskal) ได้เสนอวิธีการหาต้นไม้แบบทอดข้ามที่เล็กที่สุดเมื่อปี ค.ศ. 1956 ซึ่งมีขั้นตอนการหาง่าย ๆ ดังนี้

1.1 เลือกเส้นเชื่อมสั้นที่สุดมาหนึ่งเส้น ให้เส้นเชื่อมดังกล่าวเป็นกิ่งหนึ่งของต้นไม้แบบทอดข้ามที่กำลังหา

1.2 เลือกเส้นเชื่อมที่สั้นที่สุดซึ่งยังไม่ได้ถูกเลือก และไม่ก่อให้เกิดวงจรเมื่อรวมเส้นเชื่อมนี้เข้าในต้นไม้แบบทอดข้ามที่กำลังหา

1.3 เลือกเส้นเชื่อมต่อไปตามเงื่อนไขในขั้นตอนที่ 1.2 จนกว่าจะมีเส้นเชื่อมถูกเลือกเป็นจำนวน n-1 เส้น โดยที่ n คือจำนวนจุดในข่ายงาน

2. อัลกอริทึมของพริม

ในปี ค.ศ.1959 โรเบิร์ต พริม (Robert Prim) ได้เสนอวิธีการหาต้นไม้แบบทอดข้ามเล็กที่สุดอีกวิธีหนึ่งดังนี้

2.1 เลือกเส้นเชื่อมสั้นที่สุดมาหนึ่งเส้น ให้เส้นเชื่อมดังกล่าวเป็นกิ่งหนึ่งของต้นไม้แบบทอดข้ามที่กำลังหา

2.2 เลือกเส้นเชื่อมสั้นที่สุดที่ยังไม่ได้ถูกเลือก ซึ่งต่อกับจุดของกิ่งในต้นไม้แบบทอดข้ามที่หาได้ก่อนหน้านี้ และไม่ก่อให้เกิดวงจรเมื่อรวมเส้นเชื่อมนี้เข้าในต้นไม้แบบทอดข้ามที่กำลังหา (นั่นคือเส้นเชื่อมที่เลือกมานี้ต้องต่อกับต้นไม้ที่หาได้เพียงหนึ่งจุดเท่านั้น)

2.3 เลือกเส้นเชื่อมต่อไปตามเงื่อนไขในขั้นตอนที่ 2.2 จนกว่าจะมีเส้นเชื่อมถูกเลือกเป็น

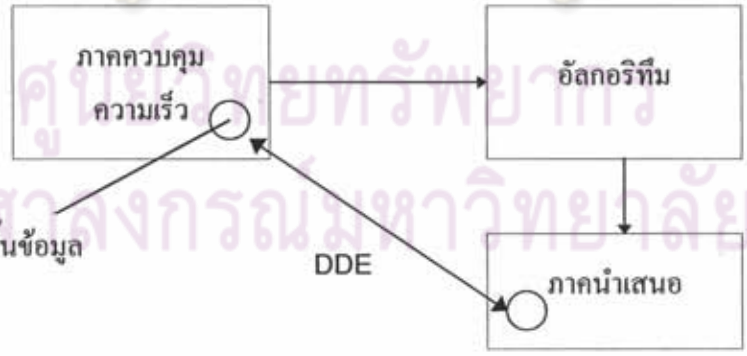
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

จำนวน n-1 เส้น โดยที่ n คือจำนวนจุดในข่ายงาน

ซอฟต์แวร์ที่นำมาใช้

ในการออกแบบและพัฒนาระบบจินตทัศน์อัลกอริทึมสำหรับปัญหาทางทฤษฎีกราฟ ได้ทำภายใต้วินโดว์ ซึ่งผู้วิจัยได้เล็งเห็นว่า โปรแกรมวินโดว์มีคุณสมบัติในการนำมาใช้สำหรับการออกแบบและพัฒนาที่เหมาะสม ซึ่งมีคุณสมบัติดังต่อไปนี้

1. เป็นโปรแกรมที่มีผู้นิยมใช้เป็นจำนวนมาก
2. มีลักษณะเป็นตัวประสานงานกับผู้ใช้แบบกราฟิก (Graphic User Interface:GUI) กล่าวคือใช้รูปภาพในการสื่อความหมาย เช่น ใช้สัญรูป(Icon) แทนชื่อโปรแกรม
3. ง่ายต่อการเรียนรู้
4. สนับสนุนการทำงานแบบหลายภารกิจ (Multitasking) ทำให้ผู้ใช้สามารถเห็นการทำงานของอัลกอริทึมได้หลายมุมมอง
5. สามารถใช้การแลกเปลี่ยนข้อมูลแบบพลวัต (DDE) โดยที่เมื่อชิ้นข้อมูลในภาคควบคุมความเร็ว มีการเปลี่ยนแปลงจะส่งผลให้ชิ้นข้อมูลที่เชื่อมต่อกับภาคนำเสนอมีการเปลี่ยนแปลงไปด้วย และภาคนำเสนอจะนำค่าของชิ้นข้อมูลที่มีการเปลี่ยนแปลงไปใช้สำหรับการเพิ่มหรือลดความเร็วในการแสดงผลการ ทำงานของอัลกอริทึม ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 แสดงการใช้ DDE ในการปรับความเร็วในการแสดงผลการทำงานของอัลกอริทึม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สำหรับในส่วนของการพัฒนาโปรแกรม ได้ใช้ภาษาวิชวลเบสิก (Visual Basic) สำหรับวินโดว์ เนื่องจากเป็นภาษาที่สามารถเรียนรู้ได้เร็ว เข้าใจง่าย ขั้นตอนในการเขียนโปรแกรมไม่ซับซ้อนซึ่งทำให้ประหยัดเวลา นอกจากนี้หนังสืออ้างอิงยังมีมากทำให้สะดวกในการค้นคว้าและในการแก้ปัญหาต่าง ๆ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย