|  |  |  |
| --- | --- | --- |
|  | **ใบความรู้ที่ 3** | จำนวนชั่วโมง  5 ชั่วโมง |
| **ชื่อวิชา** โครงสร้างข้อมูลและอัลกอริทึม **รหัสวิชา** 30901-2001 (Data Structures and Algorithms) |
| **ชื่อหน่วย** โครงสร้างข้อมูลแบบ ลิงค์ลิส (Linked Lists) |  |
|  |  |

|  |
| --- |
| **1.จุดประสงค์การเรียนรู้**  1. อธิบายโครงสร้างข้อมูลแบบลิงค์ลิสต์ได้  2. อธิบายลิงค์ลิสต์เดี่ยวได้  3. เขียนคำสั่งการสร้างโหนดได้  4. อธิบายหลักการการแทรกโหนดในลิงค์ลิสต์ได้  5. อธิบายการท่องเข้าไปในลิงค์ลิสต์ได้  6. อธิบายหลักการลบโหนดออกจากลิงค์ลิสต์ได้  7. อธิบายหลักการของลิงค์ลิสต์วงกลมได้  8. อธิบายหลักการของลิงค์ลิสต์คู่ได้  9. บอกความหมายของศัพท์ที่ใช้ในโครงสร้างข้อมูลแบบลิงค์ลิสต์ได้  **2. เนื้อหาสาระ**  **2.1 โครงสร้างข้อมูลแบบลิงค์ลิสต์**  โครงสร้างข้อมูลแบบลิงค์ลิสต์ เป็นการนำเอาข้อมูลแต่ละรายการมาจัดเรียงต่อกันและมีการเชื่อมโยงกันด้วยพอยน์เตอร์ ข้อมูลแต่ละรายการในลิงค์ลิสต์จะเรียกว่า โหนด (Node) แต่ละโหนดจะประกอบด้วย 2 ฟิลด์ คือ  1. data ป็นส่วนที่ใช้เก็บข้อมูล  2. Link หรือ Pointer เป็นส่วนที่เก็บแอดเดรส (Address) หรือตำแหน่งของข้อมูลตัวถัดไปในหน่วยความจำหลัก  ในการเชื่อมโยงไปยังส่วนอื่นๆ ของลิงค์ลิสต์นั้น จะใช้รูปแบบของพอยน์เตอร์เป็นตัวชี้ไปยังโหนดต่อๆ ไป ดังนั้น โครงสร้างข้อมูลแบบลิงค์ลิสต์จึงมีรูปแบบการจัดเก็บข้อมูลที่ประกอบไปด้วยโหนดหลายๆ โหนดมาจัดเรียงต่อกันและทำการเชื่อมโยงโหนดด้วยพอยน์เตอร์ ทำให้โครงสร้างข้อมูลแบบลิงค์ลิสต์มีลักษณะ    **ตัวอย่าง**  การประกาศโครงสร้างข้อมูลแบบลิงค์ลิสต์ มักจะประกาศเป็นตัวแปรแบบ structure ที่ ประกอบไปด้วยข้อมูล 2 ส่วน คือ data (ซึ่งอาจประกอบด้วยข้อมูลหลายตัวก็ได้ ) และ next ดังตัวอย่าง  ที่มา https://www.ict.up.ac.th/  **2.2 ลักษณะของลิงค์ลิสต์**  1.**เป็นโครงสร้างข้อมูลชนิดไม่เป็นเชิงเส้น (Non Linear Structure)** คือ จัดเก็บข้อมูลในหน่วยความจำแบบไม่ต่อเนื่องกัน การเขียนโปรแกรมจะใช้พอยเตอร์ (Pointer)  2.**ไม่จำเป็นต้องระบุจำนวนของข้อมูลที่จัดเก็บ** เนื่องจากสามารถขอหน่วยความจำใหม่ได้ เมื่อต้องการจัดเก็บข้อมูลเพิ่ม จำทำให้ไม่ต้องระบุจำนวนข้อมูลที่จะจัดเก็บไว้ตั้งแต่ตอนกำหนดตัวแปร  3.**ขนาดของหน่วยความจำที่ใช้เท่ากับข้อมูลที่จัดเก็บ** คือ หน่วยความจำที่ใช้งานจะพอดีกับข้อมูลเพราะไม่ได้ระบุขนาดไว้ก่อนจำทำให้ไม่มีหน่วยความจำที่จองไว้เหลือเหมือนการใช้อาร์เรย์  4.**ต้องมีพอยเตอร์ชี้โหนดแรก**หากไม่มีพอยเตอร์ที่จำตำแหน่งที่อยู่ของโหนดแรกแล้วก็จะไม่สามารถเข้าถึงข้อมูลที่จัดเก็บในโหนดต่างๆ ได้  **2.3 ประเภทของลิงค์ลิสต์**  ลิงค์ลิสต์ทางเดียว  ลิงค์ลิสต์สองทาง  **2.4 ลักษณะของการเก็บข้อมูลและเชื่อมโยงโหนดอื่น ๆ**             ลักษณะของการเก็บข้อมูลและเชื่อมโยงโหนดอื่น ๆ ของลิงค์ลิสต์ เริ่มจากจุดเริ่มต้นของโครงสร้าง (Start Pointer) ซึ่งเป็นตัวแปรที่ทำหน้าที่เก็บตำแหน่งของข้อมูลที่อยู่โหนดแรกในโครงสร้างชี้ไปยังโครงสร้างข้อมูลชุดถัดไป และในโครงสร้างชุดดังกล่าวนี้ก็มี Pointer ชี้ไปยังโครงสร้างข้อมูลอื่น ๆ ต่อไปในลักษณะเดียว ส่วน Pointer ในโหนดสุดท้ายจะเก็บค่า NULL (ค่าว่าง) บางครั้งแทนตำแหน่งสุดท้ายในโครงสร้างด้วยสัญลักษณ์ทางไฟฟ้า เรียกว่า ground symbol เป็นการแสดงตำแหน่งสุดท้ายในโครงสร้าง หรือ          **ที่มา** [**http://ting-tom.blogspot.com**](http://ting-tom.blogspot.com)  **2.4 การดำเนินการกับลิงค์ลิสต์**  การดำเนินการบนลิงค์ลิสต์จะมี 3 หลักๆ ได้แก่  1. การเพิ่มโหนด (Insert Node)  2.การลบโหนด (Delete Node)  3. การค้นหาโหนด (Search Node)  **การเข้าถึงข้อมูลภายในโครงสร้างลิงค์ลิสต์**  การเข้าถึงข้อมูลภายในโครงสร้างลิงค์ลิสต์ จะต้องอาศัยพอยน์เตอร์ เป็นตัวเข้าไปในโครงสร้าง สมมติให้พอยน์เตอร์ดังกล่าว คือ PTR  และทำหน้าที่ชี้ตำแหน่งแอดเดรสของโหนดในโครงสร้าง เมื่อต้องการไปยังโหนดถัดไปก็ให้ทำการเลื่อนตำแหน่งของพอยน์เตอร์  โดยตำแหน่งของโหนดถัดไปได้จากส่วนของ  LINK  ในโหนดปัจจุบัน    **ที่มา** **http://ting-tom.blogspot.com/**  **ขั้นตอนการเข้าถึงข้อมูลในโครงสร้าง**  การเข้าถึงในโครงสร้างเรียกว่า ***การทำ Traversing*** มีขั้นตอนดังต่อไปนี้กำหนดให้ DATA เป็นโครงสร้างข้อมูลลิงค์ลิสต์   และพอยน์เตอร์  PTR ทำหน้าที่ชี้โหนดที่กำลังดำเนินการ  Process  อยู่ในขณะนั้น  (Current Node)             1. กำหนดค่าเริ่มต้นให้กับพอยน์เตอร์  PTR.             2. การวนรอบดำเนินการ   Process  ข้อมูล             3. Apply Process to DATA [PTR]             4. เปลี่ยนค่าพอยน์เตอร์  PTR ให้ชี้โหนดถัดไป             5. เสร็จสิ้นขั้นตอน  **Storage Pool**                  Storage Pool หรือ Free List  หมายถึง  เนื้อที่ว่างในหน่วยความจำ มีลักษณะเป็นโหนดเก็บอยู่ในโครงสร้างข้อมูลลิงค์ลิสต์  หรืออาจเรียกได้ว่าเป็น Free  Stack  ลักษณะการดำเนินการเหมือนกับโครงสร้างข้อมูลสแต็ก    เมื่อมีการเพิ่มสมาชิกใหม่ในโครงสร้างข้อมูลลิงค์ลิสต์จะนำโหนดว่าง 1 โหนดออกมาจาก Free List (เป็นโหนดแรกใน Free List) จากนั้นใส่ข้อมูลลงไปในส่วนของ Data Field หลังจากนั้น นำโหนดดังกล่าวเชื่อมโยงเข้าไปไว้ในโครงสร้างข้อมูลที่ต้องการ และหากมีการลบสมาชิกตัวใดตัวหนึ่งออกจากโครงสร้างจะต้องนำโหนดที่ถูกลบนี้ใส่คืนใน Free List ไว้เป็นโหนดแรกใน  Free  List  เสมอ    **ที่มา http://ting-tom.blogspot.com/**   1. **การเพิ่มข้อมูลในโครงสร้าง**               เมื่อกำหนดโครงสร้างข้อมูลเรียบร้อยแล้ว ก็สามารถทำการเพิ่มข้อมูลในโครงสร้างได้   โดยการขอโหนดว่างจาก   free  list  และนำมาเชื่อมโยงกับรายการข้อมูลที่มีอยู่เดิมในโครงสร้างตรงตำแหน่งที่ต้องการ           การเพิ่มข้อมูลในโครงสร้างข้อมูลลิงค์ลิสต์ อาจเกิดในลักษณะที่ต่างกัน ซึ่งสรุปได้เป็น 3 ลักษณะ  คือ        1. การเพิ่มข้อมูลที่จุดเริ่มต้นของโครงสร้าง        2. การเพิ่มข้อมูลต่อจากโหนดที่กำหนด        3. การเพิ่มข้อมูลที่จุดสุดท้ายของโครงสร้าง  **การเพิ่มข้อมูลที่จุดเริ่มต้นของโครงสร้าง**              เป็นการเพิ่มโหนดของข้อมูลไปยังตำแหน่งแรกของโครงสร้างลิงค์ลิสต์ โดยการเปลี่ยนค่าเริ่มต้นให้ชี้ไปยังตำแหน่งของโหนดใหม่ (NEW Node) ที่สร้างขึ้น และให้ Pointer ของโหนดใหม่ชี้ไปยังตำแหน่งเริ่มต้นเดิมแทน    **การเพิ่มข้อมูลเป็นโหนดสุดท้ายของโครงสร้าง**                   เป็นการนำโหนดข้อมูลใหม่มาต่อยังตำแหน่งท้ายสุดของโครงสร้าง (Pointer ของโหนดสุดท้าย   มีค่าเป็น  NULL)  โดยการกำหนดให้ Pointer ของโหนดข้อมูลสุดท้าย ชี้ไปยังโหนดใหม่ และให้Pointer ของ โหนดใหม่มีค่าเป็น NULL แทน  **2).การลบข้อมูลจากโครงสร้าง**                   การลบข้อมูลจากโครงสร้าง หมายถึง การดึงเอาโหนดที่ต้องการลบออกจากลิงค์ลิสต์ชุดเดิม   ดังนั้น การเปลี่ยนแปลงที่เกิดขึ้นคือ  การเปลี่ยนค่าพอยน์เตอร์และเมื่อทำการลบข้อมูลออกจากโครงสร้างแล้วจะต้องคืนโหนดที่ถูกลบให้กับ Storage Pool เพื่อที่จะได้สามารถนำหน่วยความจำส่วนนั้นไปใช้งานต่อไป                  การลบข้อมูลออกจากโครงสร้างลิงค์ลิสต์ เกิดขึ้นได้หลายลักษณะสรุปได้ดังนี้              1. การลบโหนดแรก              2. การลบโหนดที่อยู่หลังโหนดที่กำหนด              3. การลบโหนดสุดท้าย  **ขั้นตอนการลบโหนดมีดังนี้**                  1. เก็บค่าตำแหน่งและค่าของ Pointer ของโหนดที่ต้องการลบ                  2. กำหนดค่าของ Pointer ของโหนดที่ต้องการลบ ไปยังโหนดก่อนหน้านั้น                  3. กำหนดตำแหน่งของโหนดที่ต้องการลบคืนกลับไปยัง Storage Pool      3.) การค้นหาโหนด (Search Node)  การท่องเข้าไปในลิงค์ลิสต์ หมายถึง การเข้าถึงโหนดทีละโหนดที่อยู่ในลิงค์ลิสต์นั้นโดยเริ่มต้นตั้งแต่โหนดแรกไปจนถึงโหนดสุดท้าย โดยทั่วไปเมื่อมีการจัดการกับลิงค์ลิสต์เรียบร้อยแล้ว หากต้องการพิมพ์ค่าที่เก็บในลิงค์ลิสต์ทั้งหมดออกมา เราสามารถท่องเข้าไปในลิงค์ลิสต์เพื่อพิมพ์ค่าที่เก็บในโหนดทีละโหนดได้ หรือถ้าจะหาว่ามีข่าวสารที่ต้องการอยู่ในลิงค์ลิสต์นี้หรือไม่ เราก็สามารถรู้ได้โดยการค้นหาแบบลำดับ (Sequential Search) โดยตั้งต้นจาก head และค้นหาไปเรื่อยๆ ตามลำดับ หรือเป็นการท่องเข้าไปเพื่อหาผลรวมของคะแนนดิบของนักศึกษาทั้งหมดแล้วนำมาคิดเป็นคะแนนเฉลี่ย เป็นต้น  ในการท่องเข้าไปในลิงค์ลิสต์จะต้องทำการกำหนดพอยน์เตอร์ขึ้นมาเพื่อท่องเข้าไป ซึ่งเราจะไม่ใช้ head เนื่องจากถ้าเลื่อน head แล้วจะทำให้กลับไปที่ต้นลิงค์ลิสต์ไม่ได้ และจะมีการกำหนดลูปเพื่อให้พอยน์เตอร์เคลื่อนที่ไปยังโหนดถัดไป ซึ่งสามารถเขียนคำสั่งการท่องเข้าไปในลิงค์ลิสต์ได้ดังนี้ |
|  |