

## Lab11 DFS&HEAP

1. กำหนดกราฟดังภาพ จงทำการเก็บกราฟด้วย adj list และ search graph ด้วย DFS

=====MENU=====

1) Insert Graph //insert แล้วให้เก็บข้อมูลลงใน adj list

2) DFS

3) Exit

Please choose >

กรณีที่เลือกข้อ 1 ให้ทำการ insert ข้อมูล

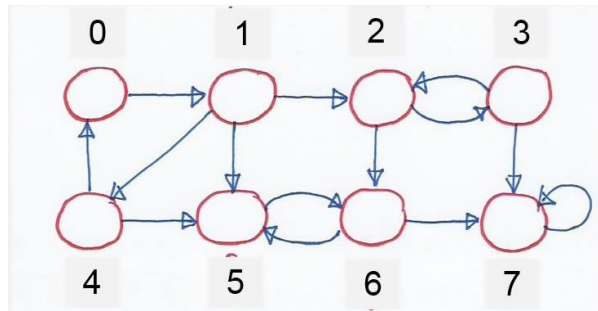
#0 : 1 -1

#1 : 2 4 5 -1

#2 : 3 6 -1

...

#7 : 7 -1



กรณีที่เลือกข้อ 2 ให้ผู้ใช้ใส่หมายเลขโหนดต้นทาง

สมมติผู้ใช้ใส่โหนด 1 เป็นโหนดต้นทาง ให้ทำ DFS

- 2.1 แสดงจำนวน DFS Trees
- 2.2 แสดงข้อมูลใน array d, f และ pred

Input source vertex : 1

Trees = 1 //มี 1 ต้น

d : 13 1 2 3 12 8 7 4

f : 14 16 11 6 15 9 10 5

pred : 4 -1 1 2 1 6 2 0

2. กำหนดกราฟดังข้อที่ 1 จงทำการ Decompose strongly connected graph ด้วย DFS

=====MENU=====

- 1) Insert Graph //insert แล้วให้เก็บข้อมูลลงใน adj list
- 2) DFS
- 3) Transpose graph
- 4) Decompose strongly connected graph //ทำการแตกกราฟเป็นกลุ่ม strongly connected
- 5) Exit

Please choose >

กรณีที่เลือกข้อ 1 และ 2 ให้ทำเหมือนข้อที่แล้ว

ถ้าเลือกข้อ 3 ให้แสดงข้อมูลใน adj list และ adjacency list ที่ traspose แล้ว

แสดงผลลัพธ์โดยพิมพ์ข้อมูลใน adjacency list ดังนี้

Adjacency list

0 : 1

1 : 2 4 5

2 : 3 6

3 : 2 7

4 : 0 5

5 : 6

6 : 5 7

7 : 7

Adacency list transpose

0 : 4

1 : 0

2 : 1 3

3 : 2

4 : 1

5 : 4 6

6 : 2 5

7 : 3 6 7

ถ้าเลือกข้อ 4 ให้แสดง แสดงข้อมูลแต่ละ tree (ที่แตกออกมาเป็น group แล้ว) ดังภาพ

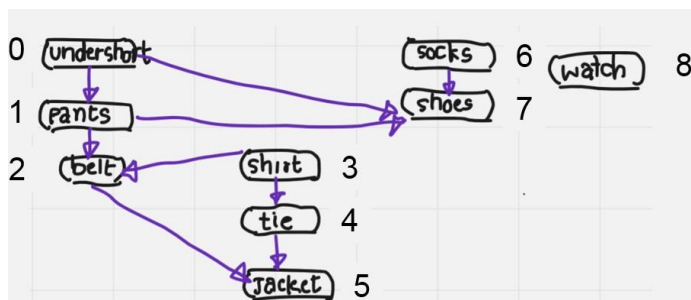
1 ( 0 ( 1 ( 4 4 ) 1 ) 0 )

2 ( 2 ( 3 3 ) 2 )

3 ( 6 ( 5 5 ) 6 )

4 ( 7 7 )

3. Topological sort ต้องการแต่งตั้งไปทำงาน โดยลำดับก่อนหลังที่สามารถจะแต่งตั้งได้กำหนดดังกราฟด้านล่าง ให้ทำงานตามตาราง



ให้ทำการ DFS โดยให้โหนดเริ่มต้นเป็นโหนดที่หมายเลขที่น้อยที่สุดที่สามารถทำ topological sort ได้ ในที่นี้คือโหนด 0

=====

MENU

=====

1. Insert Graph
2. Topological sort
3. Exit

Please choose >

กรณีข้อ 2 ให้ทำการ เรียงข้อมูลโดยใช้ Topological sort เริ่ม DFS จากโหนดที่ 0

จากนั้น ให้พิมพ์ข้อมูลเรียงลำดับการแต่งตัวที่เก็บอยู่ใน linked list ตามหมายเลขโหนด

8 6 3 4 ... 5

ถ้าใครสามารถเรียงข้อมูลได้โดยพิมพ์เป็นข้อความจะได้คะแนนเต็ม 10 ถ้าเป็นตัวเลขได้ 9

watch socks .... jacket