

**ภาคผนวก**

## เฉลยคำถามท้ายบทที่ 1

1. Measurement, collection, transcription, validation, organization, storage, aggregation, update, retrieval, protection

Most expensive: collection and transcription, because this is where human labor commonly is used extensively

2. 20, 2, -6, 0 และอื่น ๆ

3. true, false

4. (a) False (b) false (c) true (d) true  
(e) true (f) true (g) false (h) false  
(i) true (j) true (l) true (l) false

5. (a) 5 (b) 6 (c) 'MAGICPIE'  
(d) 'EEL' (e) 'MAGPIE' (f) 'MAG WHEEL'  
(g) 'PIE' (h) 'PIE' (l) 'PILE'  
(j) 'MIC' (k) 'EELPIE'

6. -7 = 1000

-6 = 1001

-5 = 1010

-4 = 1011

-3 = 1100

-2 = 1101

-1 = 1110

+0 = 0000

+1 = 0001

+2 = 0010

+3 = 0011

+4 = 0100

+5 = 0101

+6 = 0110

+7 = 0111

7. (a) 21

(b) 201

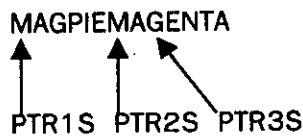
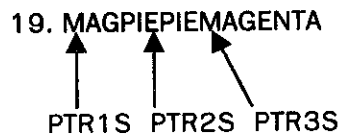
8. (a) 32 ( $R = 2^5$ )  
 (b) 256 ( $R = 2^8$ )

9. N

10. N

11. (a) +16, -15  
 (b) +128, -127  
 (c)  $2^{N-1}$ , ( $2^{N-2} - 1$ )

12. (a) +15, -15  
 (b) +127, -127  
 (c)  $2^{N-1}$ ,  $-(2^{N-2} - 1)$



20. Overlaying string conserves storage but can greatly complicate processing when the values of string variables are change.

21. Packing strings conserves storage but complicates processing when the values of string variables are modified.

22.(A) and (B) both allow overlaying of strings

(C) does not

(C) requires a smaller table than does(A) or (b)

(A) and (B) both require some processing to find the end of a string.

## เฉลยแบบฝึกหัดบทที่ 2

1. 70 ตัว
2.  $(B-A+1) * (D-C+1)$
3.  $N: 3(N+1)$
4. 16
5. (a) 1911 (b) 8578
6.  $B+ 318*S$
7. ถ้าเก็บที่สระแถว  
 $(i-B) * (A-B+1) * n + (j-D) * n$   
 ถ้าเก็บที่สระสดมภ์  $(j-D) * (C-D+1)*n+(i-B)*n$
8. (a)  $(B-A+1) * (D-C+1) *(F-E+1)$   
 (b) correct answer depends upon the order of linearization of the there dimensions
9. แถวลำดับสพาชเริ่มต้น :

row	column	value
1	3	2
2	1	2
2	5	3
3	4	9
3	6	1
4	2	16

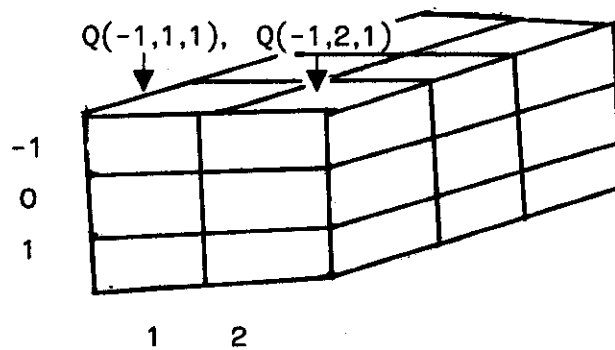
แถวลำดับ transpose:

row	column	value
3	1	1
1	2	2
5	2	3
4	3	9
6	3	1

10. (a) 18 ตัว

(b)  $Q(-1,1,1)$  ,  $Q(-1,1,2)$ ,  $Q(-1,1,3)$ ,  $Q(-1,2,1)$   
 $Q(-1,2,2)$  ,  $Q(-1,2,3)$ ,  $Q(0,1,1)$ ,  $Q(0,1,2)$   
 $Q(0,1,3)$  ,  $Q(0,2,1)$ ,  $Q(0,2,2)$ ,  $Q(0,2,3)$   
 $Q(1,1,1)$  ,  $Q(1,1,2)$  ,  $Q(1,1,3)$  ,  $Q(1,2,1)$ ,  
 $Q(1,2,2)$ ,  $Q(1,2,3)$

(c)  $Q(-1,1,1)$  ,  $Q(0,1,1)$ ,  $Q(1,1,1)$ ,  $Q(-1,2,1)$   
 $Q(0,2,1)$  ,  $Q(1,2,1)$ ,  $Q(-1,1,2)$ ,  $Q(0,1,2)$   
 $Q(1,1,1)$  ,  $Q(-1,2,2)$ ,  $Q(0,2,2)$ ,  $Q(1,1,2)$   
 $Q(-1,1,3)$  ,  $Q(0,1,3)$  ,  $Q(1,1,3)$  ,  $Q(-1,2,1)$ ,  
 $Q(0,2,3)$ ,  $Q(1,2,3)$



จำนวนสมาชิกในแต่ละมิติ =  $u-l+1$

มิติที่ 1 มีสมาชิก =  $1-(-1)+1 = 3$  ตัว

มิติที่ 2 มีสมาชิก =  $2-(1)+1 = 2$  ตัว

มิติที่ 3 มีสมาชิก =  $3-(1)+1 = 3$  ตัว

เพราะฉะนั้น Q มีสมาชิกทั้งหมด =  $3 \times 2 \times 3$  ตัว  
= 18 ตัว

เฉลยแบบฝึกหัดบทที่ 6 (รายการโยง)

1. FIRST = 2  
AVAIL = 3

	INFO	NEXT
1	PEACH	0
2	APPLE	4
3		0
4	CHERRY	5
5	GRAPE	1

2. FIRST = 4  
AVAIL = 7

	INFO	NEXT
1	G	3
2	D	5
3	J	0
4	B	2
5	E	6
6	F	1
7		9
8		9
9		1

3. FIRST = 2  
AVAIL = 2

	INFO	NEXT
1	G	3
2	D	5
3	J	9
4	B	7
5	E	6
6	F	1
7	C	2
8	A	4
9	Z	0

4. อัลกอริทึมนี้ต้องทำแบบลำดับผ่านโหนดต่าง ๆ ของรายการโยงเดี่ยวนับโหนดจนกระทั่งพบโหนดที่  $j$  ให้ระมัดระวังว่าในรายการนั้นอาจมีจำนวนโหนดน้อยกว่า  $j$  ตัว
5. รายการ  $T$  สามารถนำไปใส่ที่ตอนต้นหรือตอนจบของรายการว่าง (หรือที่ใด ๆ ก็ได้ระหว่าง สองตำแหน่งนี้)

การใส่ที่ตอนต้นสามารถกระทำได้ด้วยโปรซีเจอร์ต่อไปนี้ :

- (a) หาโหนดตัวสุดท้ายในรายการ  $T$  จากนั้นเปลี่ยนพอยน์เตอร์ Next ของโหนดตัวสุดท้ายให้ชี้ไปยัง โหนดตัวแรก ในรายการว่าง
- (b) หาโหนดตัวสุดท้ายในรายการว่าง จากนั้นเปลี่ยนพอยน์เตอร์ Next ของโหนดตัวนี้ให้ชี้ไปยังโหนดตัวแรกในรายการ  $T$
- (c) Avail :=  $t$

6. P

7. (a) เอาเนื้อที่ว่างให้กับโหนดตัวใหม่และให้ NEW ชี้ไปยังโหนดตัวนี้

- (b) Info(NEW) := '1234'
- (c) Next(New) := Next(x)
- (d) Next(x) := New

8. (a) a := Prior(x)

- (b) Prior(X) := S
- (c) Prior(S) := Q
- (d) Next(Q) := S
- (e) Next(S) := X

9. รายการโยงเดี่ยว

- (a) SI := S
- (b) Q := Next(S)
- (c) Do until Q = Null
- R := Next(Q)
- Next(Q) := S
- S := Q
- Q := R
- Next(SI) := Null

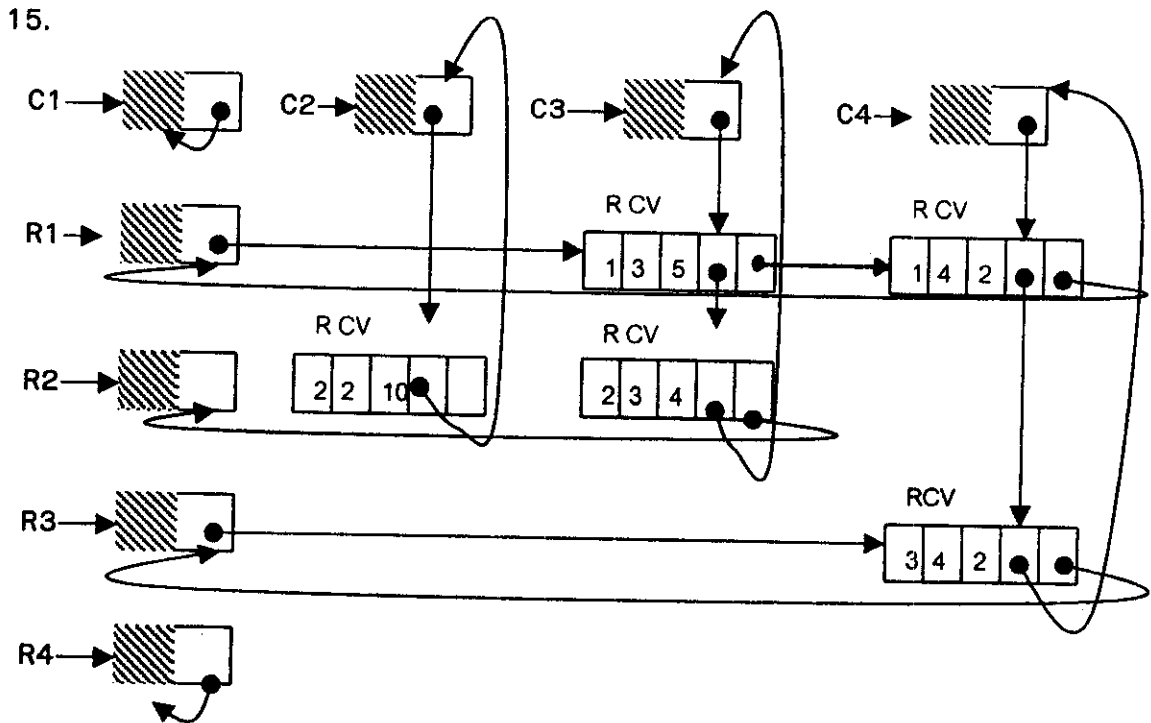
รายการโยงคู่

- (a) SI := S

- (b)  $Q := \text{Next}(s)$
- (c) Do until  $Q = \text{Null}$ 
  - $R := \text{Next}(Q)$
  - $\text{Next}(Q) := S$
  - $\text{Prior}(Q) := R$
  - $S := Q$
  - $Q := R$
- $\text{Next}(S) := \text{Null}$

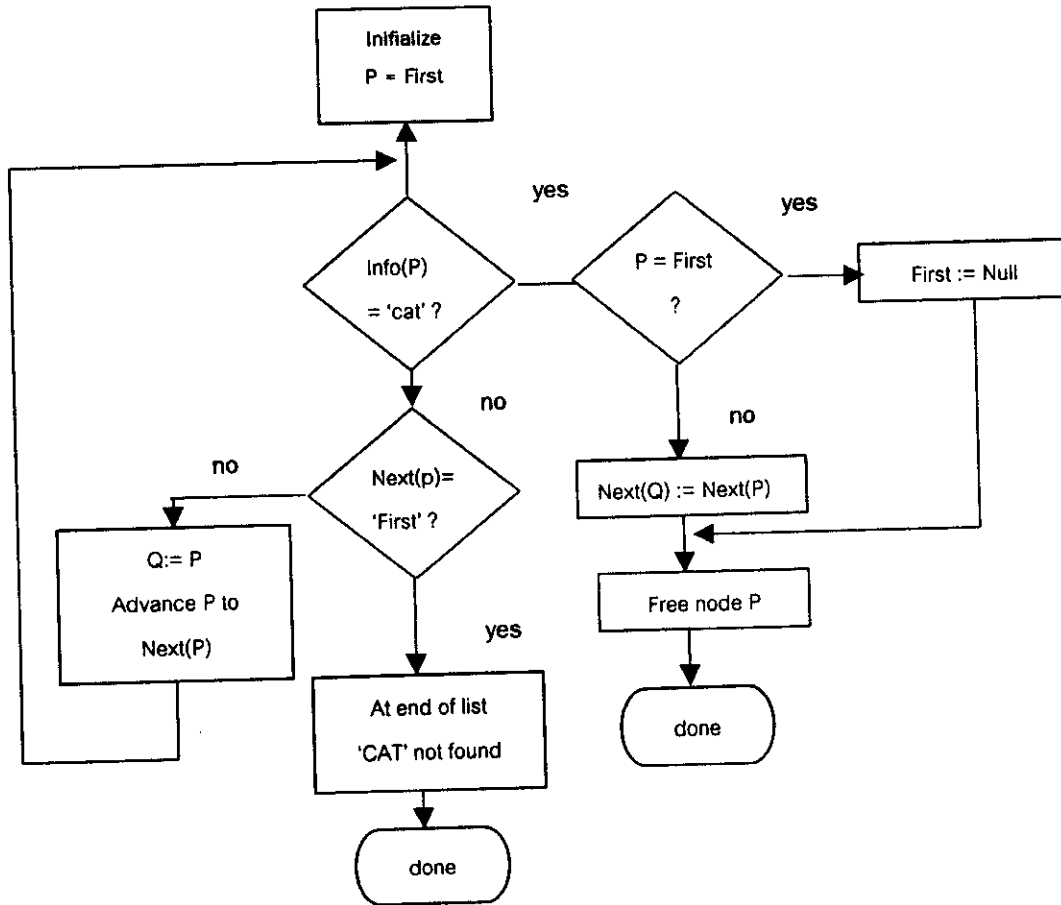
10. อัลกอริทึมเอาโหนดหนึ่งตัวจาก Avail queue เหมือนกับอัลกอริทึมเอาโหนดหนึ่งตัวจาก Avail stack อัลกอริทึมคืนโหนด แล้วใส่เพิ่มให้กับ Avail queue ทำให้ง่ายขึ้นโดยพร้อมที่จะเข้าถึงทางของแถวคอย สิ่งนี้ปกติ ทำโดยเค็พอยน์เตอร์ที่ tail

- 11. อย่างน้อยที่สุด 2 ตัว
- 12. คู่อัลกอริทึม Getnode และ Freenode สำหรับ Avail Stack
- 13. คู่อัลกอริทึม Getnode และ Freenode สำหรับ Avail queue
- 14. (a) หาโหนดตัวสุดท้ายในรายการ A และให้ LAST ชี้ที่โหนดตัวนี้  
(b)  $\text{Next}(\text{LAST}) := B$





16.



17.

(a) If Next (FIRST) = FIRST then done , empty list

(b) P := FIRST

(c) Do until Info(P) = 'BAT'

P := Next(P)

If P = FIRST then done , 'BAT' not in list

(d) Q := Next(P)

(e) If Info(Q) not = 'CAT' then done 'CAT' doesn't follow 'BAT'

(f) Get node from Avail list and make CATFIRST point to it

(g) Next(CATFIRST) := Q

(h) Next(P) := FIRST

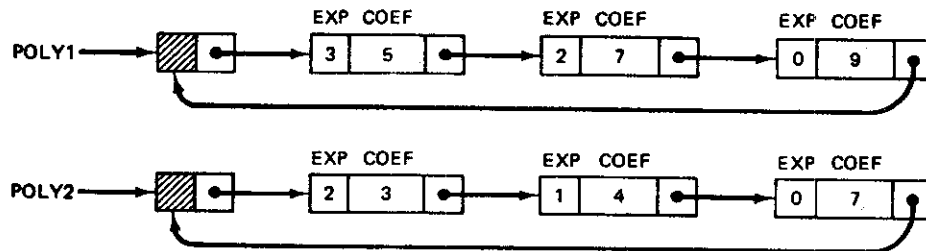
(i) R := Q

(j) Do until Next (R) = FIRST

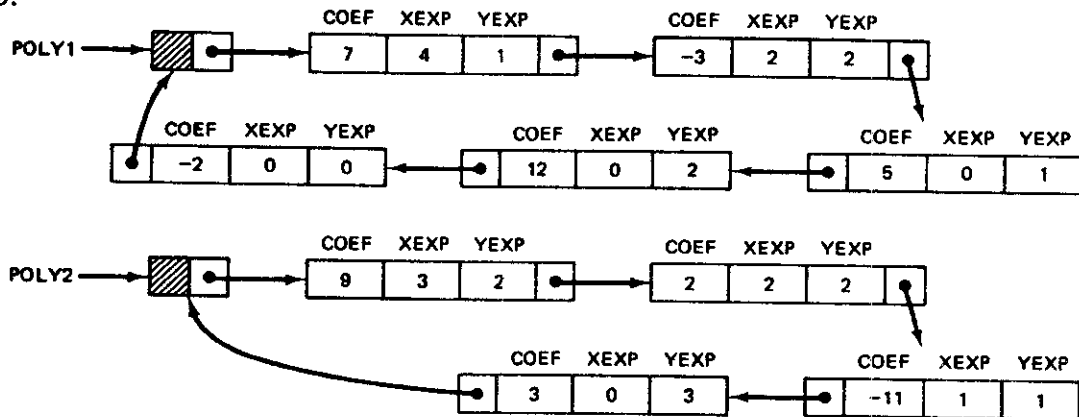
R := Next ( R )

Next ( R ) := CATFIRST

18.



19.



20 .ผ่านโหนดต่าง ๆ ในรายการแบบลำดับ เพิ่ม counter ขณะที่ประมวลผลการนับหยุด เมื่อจบรายการ ขึ้นอยู่กับโครงสร้างของรายการการจบรายการอาจมีสัญญาณโดย Null Next pointer (กรณี non-circular list) , โดยพอยน์เตอร์ Next เท่ากับพอยน์เตอร์ที่ชี้โหนดตัวแรกของรายการ (circular list) , หรือโดยพอยน์เตอร์ Next พบโหนดตัวหน้าหน้าของรายการ

21.Final state

Author	next- Author	Title	Next title	Stock-no	Next stock-no	Prior- Stock-no	Next avail
F.Stop Fitzgerald	4	Creative. . .	10	53526	10	5	0
Linus Type	0	Newspaper. . .	4	98374	0	10	0
							11
Lowen Maner	10	The Longest	0	23764	9	6	0
Ascent Agu	6	La Marke de	2	49261	1	9	0
Dr. Frank Annstein	1	Beer Basted	9	19822	4	8	0
							3
Artie Abacus	5	Computer. . .	1	17760	6	0	0
Denn M. stretcher	2	Betsy wors. . .	8	38641	5	4	0
C. sanders	9	Finger. . .	5	73920	2	1	0
							12
							13
							14
							15
							0

Headings : First-Author = 8

First-Title = 6

First-stock - no = 8

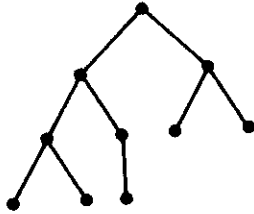
Last-stock - no = 2

Avail = 7

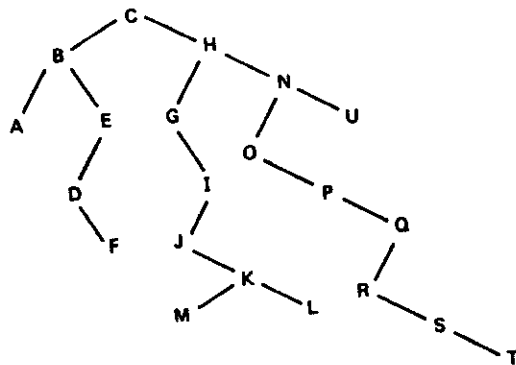
22. ข้อดี : ช่วยในการหาตอนจบของรายการแบบวงกลม , สามารถนำมาใช้เก็บสารสนเทศซึ่งอธิบายรายการ  
ข้อไม่ดี : ใช้เนื้อที่
23. ข้อดี : ช่วยในการลำดับจากโหนดใดโหนดหนึ่งไปยังอีกโหนดหนึ่งในรายการ  
ข้อไม่ดี : อาจทำให้การตรวจหา(detect) ตอนจบของรายการมีความยากมากขึ้น ถ้าโหนดสุดท้ายเก็บ null Next pointer
24. ข้อดี : ความสามารถดีขึ้นเมื่อการใส่หรือการลบโหนดใดโหนดหนึ่งออกจากรายการ โดยลดจำนวนโหนดซึ่งต้องมีการสัมผัสความสามารถดีขึ้นเมื่อการหาโหนดตัวหน้าโหนดใดโหนดหนึ่งในรายการ  
ข้อไม่ดี : ในแต่ละโหนดใช้เนื้อที่สำหรับพอยน์เตอร์ที่เพิ่มขึ้นอีกหนึ่ง
25. (a) อัลกอริทึมลบโหนดออกจากรายการโยงคู่ สัมผัสต้องโหนด 3 ตัว  
(b) อัลกอริทึมใส่โหนดในรายการโยงคู่ต้องสัมผัสโหนด 3 ตัว  
(c) แต่ละสถานะการณ์ต้องสัมผัสโหนดที่เพิ่มใน Avail list

**เฉลยแบบฝึกหัดบทที่ 8**

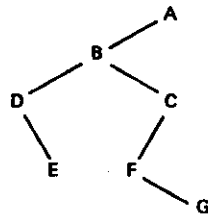
3. ไม่ได้, โดยทฤษฎีของ equivalence
4. ไม่ได้, เพราะว่าต้นไม้ส่วนย่อยแต่ละชุดต้องมีโหนดราก
5.  $2^l$
6.  $2^l$  for  $l < K-1$  และค่ามากที่สุดของ  $2^l$  for  $l = K-1$
7. เป็นไปไม่ได้ ; ; 4 ; 10



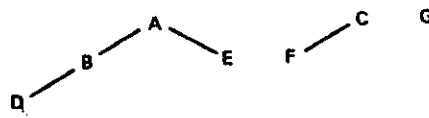
8. โหนดในต้นไม้แบบทวิภาคมีจำนวนต้นไม้ส่วนย่อยคงที่ ในขณะที่โหนดในต้นไม้ทั่วไปอาจมีจำนวนต้นไม้ส่วนย่อยได้หลากหลาย
9. (a) ABDGLPEHMQICFJKNO  
(b) PLGDQMHIEBJNOKFCA  
(c) PLGDBHQMEIACJFNKO
10. จริง
- 11.



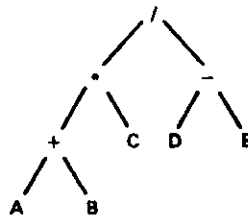
12.



13.



14.

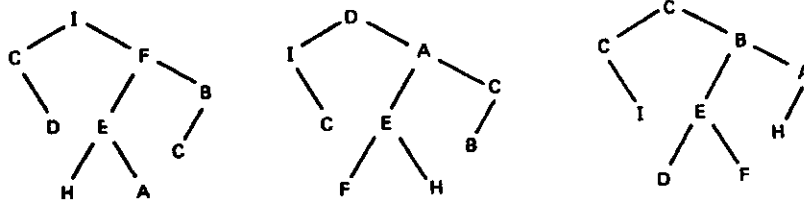


$/*+ABC-DE$

$AB+C*DE- /$

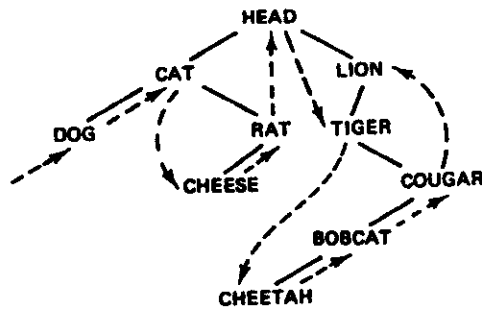
$A+B*C/D-E$

15.



16. Keep inserting the "middle" key next.

18.



19. Inorder

22. LEFT RIGHT INFO INORDER THREAD POSTORDER THREAD

23. ไม่ใช่ ; ปกติอันดับการแวะผ่านมีเพียงหนึ่งแบบเท่านั้นซึ่งเหมาะสมกับชนิดของข้อมูลซึ่งเก็บในต้นไม้ (No ; usually only one traversal order is appropriate for the type of data stored in the tree.)

35. ต้นไม้ได้ดุล : ต้นไม้ส่วนย่อยซ้ายและต้นไม้ส่วนย่อยขวามีจำนวนโหนดเท่ากัน  
ต้นไม้ความสูงได้ดุล : ต้นไม้ส่วนย่อยซ้ายและต้นไม้ส่วนย่อยขวามีความยาวทางเดินสูงสุดเกือบเท่ากัน