

1. จากฟังก์ชันการทำ Shellsort ข้างล่างนี้

( 3 คะแนน)

```

01: void ShellSort( int A[], int N )
02: {
03:     int    i, j, Increment;
04:     int    tmp;
05:
06:     for ( Increment=N/2; Increment>0; Increment /= 2 )
07:         for ( i=Increment; i<N; i++ )
08:             {
09:                 tmp = A[i];
10:                 for ( j=i; j>= Increment; j-=Increment )
11:                     if ( tmp < A[j-Increment] )
12:                         A[j] = A[j-Increment];
13:                     else
14:                         break;
15:                 A[j] = tmp;
16:             }
17: }

```

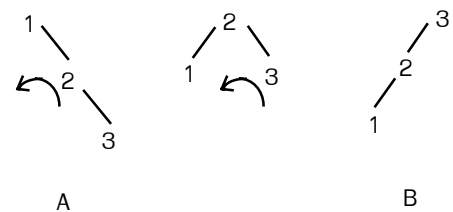
1.1) จงแก้ไขโปรแกรมให้ทำการเรียงลำดับตัวเลขใน Array A[ ] ให้เรียงจากมากไปน้อย (ระบุหมายเลขบรรทัดที่มีการเปลี่ยนแปลงกำกับด้วย)

1.2) ถ้าชุดตัวเลขใน A[ ] มีค่าดังนี้ 10, 20, 30, 5, 8, 33, 100, 25, 36, 41, 35, 67, 35, 23, 18, 39 จงบอก Increment Sequence ของค่า Increment ในโปรแกรม และพิจารณาว่า Increment Sequence ดังกล่าวเหมาะสมหรือไม่อย่างไร

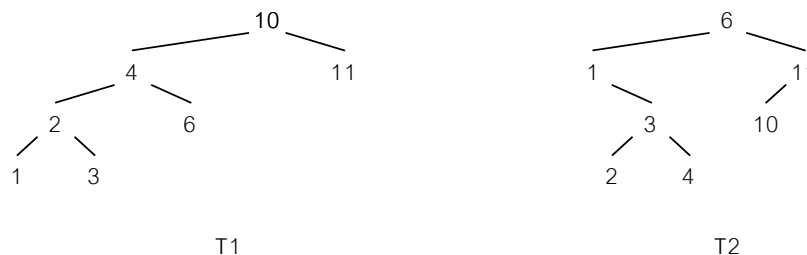
2.

( 5 คะแนน)

เราสามารถแปลงลักษณะของ binary search tree จากต้นหนึ่งไปเป็น binary search tree รูปลักษณะใดๆก็ได้ (ที่มีจำนวนข้อมูลและค่าของข้อมูลเหมือนกัน) โดยการใช้วิธีการหมุนแบบ single rotation ตัวอย่างเช่นเราสามารถแปลงต้นไม้ A ไปเป็น B ทางขวานี้ได้ (โดยการหมุนที่โหนด 2 จากนั้นหมุนที่โหนด 3 )



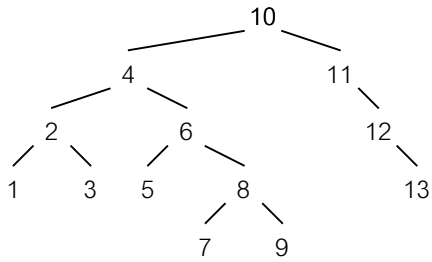
2.1) จงเสนอขั้นตอนการหมุนเพื่อแปลงต้นไม้จากต้น T1 เป็น T2 (แสดงข้างล่างนี้) วาดการเปลี่ยนแปลงของต้นไม้ประกอบด้วย



2.2) จงบรรยายอัลกอริทึมเพื่อเปลี่ยนแปลง binary search tree จากรูปแบบหนึ่งไปเป็นอีกรูปแบบหนึ่งที่กำหนดให้ใดๆ ด้วยวิธี single rotation (ไม่ต้องเขียนเป็นโปรแกรม จะเขียนก็ได้แต่ต้องอธิบายประกอบด้วย)

3. จงเขียนฟังก์ชันที่รับต้นไม้ binary search tree หนึ่งต้น เพื่อตรวจสอบว่าต้นไม้มีคุณสมบัติเป็น AVL หรือไม่ ถ้าต้นไม้ที่ทดสอบนี้เป็นต้นไม้ AVL ค่าที่คืนจากฟังก์ชันนี้ จะเป็นความสูงของต้นไม้ (กำหนดให้ต้นไม้ NULL มีความสูงเป็น -1) แต่ถ้าไม่ใช่ต้นไม้ AVL จะคืนเลข -2 (กำหนดให้ต้นไม้ที่รับมาเป็น binary search tree ซึ่ง structure ของ node ไม่มีสมาชิกที่เก็บความสูงของต้นไม้ด้วย) ( 4 คะแนน)

4. จงวาดผลลัพ์ที่ได้หลังจากมีการอ้างอิงข้อมูล 3 และ 9 ตามลำดับในต้นไม้ splay ข้างล่างนี้ ( 3 คะแนน)



5. ในกรณีของการใช้โครงสร้างแบบ separate chaining นั้นโดยทั่วไปเราจะทำการ rehash เมื่อมีการเปลี่ยนแปลงขนาดของ table สมมติว่าได้ออกแบบไว้ให้มีการขยายขนาดของ table ขึ้นสองเท่าตามด้วยการ rehash เมื่อจำนวนข้อมูลมีขนาดเป็นสองเท่าของขนาดของ table และเพื่อให้ใช้ประโยชน์ของหน่วยความจำได้คุ้มค่าๆ ก็ได้กำหนดว่าจะต้องลดขนาดของ table ลงเป็นครึ่งหนึ่งของขนาดเดิมด้วย (ตามด้วยการ rehash) เมื่อข้อมูลมีจำนวนน้อย อยากรทราบ ว่า ที่ว่าเมื่อข้อมูลมีจำนวนน้อยนั้น ควรจะเป็นสักเท่าใดเมื่อเทียบกับขนาดของ table ให้เหตุผลประกอบด้วย ( 3 คะแนน)

6. วิธีการเรียงลำดับต่อไปนี้วิธีใด (เลือกหนึ่งวิธี) ที่ไม่เหมาะสมอย่างยิ่งในการเรียงลำดับข้อมูลที่เก็บใน linked list : Insertion Sort, Shell Sort, Merge Sort, Quick Sort ให้เหตุผลประกอบสั้นๆ ด้วย ( 2 คะแนน)

7. จงเติมคำสั่งภาษา C ในกรอบต่างๆของฟังก์ชัน Find ข้างล่างนี้ เพื่อการหาข้อมูลแบบ quadratic probing (ฟังก์ชันนี้ใช้ structure เดียวกับที่เรียนมา) ( 3 คะแนน)

```
Position Find( float key, Hashtable H )
```

```
{
    Position      CurrentPos;
    int           CollisionNum;
```

```
CurrentPos = Hash( Key, H->TableSize );
while( H->TheCells[ CurrentPos ].Info != Empty &&
       H->TheCells[ CurrentPos ].Element != Key )
{
```

```
}
return CurrentPos;
}
```

8. จงเขียนฟังก์ชัน FindMin ที่คืนค่าของข้อมูลตัวที่มีค่าน้อยสุด ใน MaxHeap ที่กำหนดให้ ให้คำนึงถึงประสิทธิภาพเชิงเวลาของฟังก์ชันนี้ (นั่นคือฟังก์ชันนี้ต้องทำงานได้เร็วสุดเท่าที่จะเร็วได้) ให้ใช้ structure ของ heap ที่เรียนมา ไม่ต้องเขียน structure นั้นในคำตอบ ( 3 คะแนน)

9. จงเขียนฟังก์ชัน Find3rdMax ที่คืนค่าของข้อมูลตัวที่มีค่ามากที่สุดเป็นอันดับที่สามใน MaxHeap ที่กำหนดให้ ฟังก์ชันนี้ต้องทำงานโดยใช้เวลา  $O(1)$  ให้ใช้ structure ของ heap ที่เรียนมา ไม่ต้องเขียน structure นั้นในคำตอบ (3 คะแนน)
10. จงเขียนข้อมูลใน array ที่แทน binary heap หลังการ BuildMaxheap จากข้อมูลใน array ข้างล่างนี้ (กำหนดให้รากของ binary heap อยู่ที่ index 1 ของอาเรย์) วาดต้นไม้ระหว่างการเปลี่ยนแปลงประกอบด้วย (3 คะแนน)

1	2	3	4	5	6
6	9	10	13	16	22

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

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

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

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

11. กำหนดให้  $\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}$  คือ disjoint set 8 sets ถ้าแทน set ด้วยต้นไม้ และการ union set นั้นทำแบบ union by height จงหาลำดับการ union เซตเหล่านี้จำนวน 7 ครั้งที่ทำให้ได้ set  $\{1,2,3,4,5,6,7,8\}$  โดยต้นไม้ที่แทนเซตหลังการ union ทั้งหมดนั้น มีความสูงสูงสุด (3 คะแนน)
12. ฟังก์ชันข้างล่างนี้เรียงลำดับข้อมูลที่เก็บในอาเรย์ A ตั้งแต่ตัวที่ index Left ถึงตัวที่ index Right หรือไม่ ยกตัวอย่างประกอบคำตอบ (3 คะแนน)

```
void HaHaSort( int A[], int Left, int Right )
{
    int k;

    if ( A[Left] > A[Right] ) Swap( &A[Left], &A[Right] );
    if ( Left+1 >= Right ) return;
    k = (Right - Left + 1)/3;
    HaHaSort( A, Left, Right-k );
    HaHaSort( A, Left+k, Right );
    HaHaSort( A, Left, Right-k );
}
```

13. จงเติมคำสั่งภาษา C ในกรอบต่างๆของฟังก์ชัน Qsort ข้างล่างนี้ เพื่อการเรียงลำดับข้อมูลแบบ QuickSort ที่เก็บในอาร์เรย์ A ตั้งแต่ตัวที่ index Left ถึงตัวที่ index Right ( 4 คะแนน)

```
int Median3( int A[], int Left, int Right )
{
    int        Center = (Left+Right)/2;

    if ( A[Left  ] > A[Center] ) Swap( &A[Left  ], &A[Center] );
    if ( A[Left  ] > A[Right ] ) Swap( &A[Left  ], &A[Right ] );
    if ( A[Center] > A[Right ] ) Swap( &A[Center], &A[Right ] );
    Swap( &A[Center], &A[Right-1] );
    return( A[Right-1] );
}

void QSort( int A[], int Left, int Right )
{
    int        i, j;
    int        Pivot;

    if ( Left + Cutoff <= Right )
    {
        Pivot = Median3( A, Left, Right );

        for ( ; ; )
        {
            

            if ( i < j )
                Swap( &A[i], &A[j] );
            else
                break;
        }
        Swap( &A[i], &A[Right - 1] );

        
    }
    else
        InsertionSort( A+Left, Right-Left+1 );
}

```

14. Anagram คือคำที่ถูกสร้างขึ้นโดยการสลับลำดับของตัวอักษรของคำอีกคำหนึ่ง ตัวอย่างเช่น *วนอ อวน วนวน นวน* และ *นวน* ต่างเป็น anagram ของคำว่า *วนอ* สิ่งที่น่าสนใจคือ anagram ที่เป็นคำที่มีความหมาย (นั่นคือคำที่ปรากฏในพจนานุกรม) เช่น *อวน* เป็น anagram ที่มีความหมายของคำว่า *วนอ* จงออกแบบการจัดเก็บพจนานุกรม เพื่อใช้หาว่ามี anagram ใดบ้างของคำที่กำหนดให้ที่มีเก็บอยู่ในพจนานุกรม อย่างมีประสิทธิภาพ กำหนดให้มีคำศัพท์ที่จะจัดเก็บอยู่ประมาณหนึ่งหมื่นคำ ( 3 คะแนน)

15. ต้นไม้แดงดำ (Red-Black Tree) คือ binary search tree ที่แต่ละ node มีสีกำกับตามเงื่อนไขดังนี้

- แต่ละโหนดต้องมีสีกำกับว่าเป็นโหนดแดง หรือโหนดดำ
- กำหนดให้ รากเป็นโหนดดำเสมอ
- ลูกทุกลูกของโหนดแดง ต้องเป็นโหนดดำ
- ทุกๆทางจากรากถึง NULL ใดๆในต้นไม้ ต้องมีจำนวนโหนดที่เป็นสีดำ เท่ากันหมด

( 3 คะแนน)

15.1) จงวาดต้นไม้แดงดำที่สูง 1 ที่มีจำนวนโหนดน้อยที่สุด (ระบุสีกำกับกับโหนดต่างๆด้วย)

15.2) จงวาดต้นไม้แดงดำที่สูง 2 ที่มีจำนวนโหนดน้อยที่สุด (ระบุสีกำกับกับโหนดต่างๆด้วย)

15.3) จงวาดต้นไม้แดงดำที่สูง 3 ที่มีจำนวนโหนดน้อยที่สุด (ระบุสีกำกับกับโหนดต่างๆด้วย)

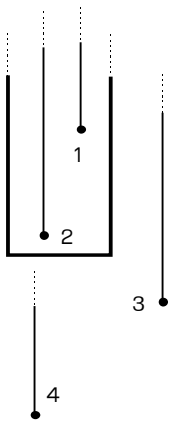
16. ปัญหาดอกไม้และแจกัน : กำหนดให้

- แจกันหนึ่งใบ คือลำดับของตัวเลขสามตัว  $(x_L, x_R, y)$  ซึ่งเป็นเส้นตรงสามเส้น  $(x_L, y)-(x_L, \infty)$ ,  $(x_R, y)-(x_R, \infty)$ , และ  $(x_L, y)-(x_R, y)$
- ดอกไม้หนึ่งดอก คือคู่อันดับ  $(x', y')$  ซึ่งเป็นเส้นตรงหนึ่งเส้น  $(x', y')-(x', \infty)$

กำหนดให้ดอกไม้  $(x', y')$  อยู่ในแจกัน  $(x_L, x_R, y)$  เมื่อ

$$x_L \leq x' \leq x_R \text{ และ } y \leq y'$$

รูปทางซ้ายนี้แสดงตัวอย่างดอกไม้ 4 ดอก และแจกันหนึ่งใบ โดยดอกไม้หมายเลข 1 และ 2 เป็นดอกไม้ที่อยู่ในแจกัน



จงออกแบบโครงสร้างข้อมูลที่จัดเก็บแจกัน และดอกไม้หลายๆดอก เพื่อให้สามารถตอบคำถาม (ได้เร็วๆ) ว่ามีดอกไม้ดอกใดอยู่ในแจกันที่กำหนดให้บ้าง

( 5 คะแนน)