Department of Computer Science and Engineering
National Sun Yat-sen University
Data Structures - Final Exam., Jan. 7, 2019

1. What are printed by the following C program? (8%)

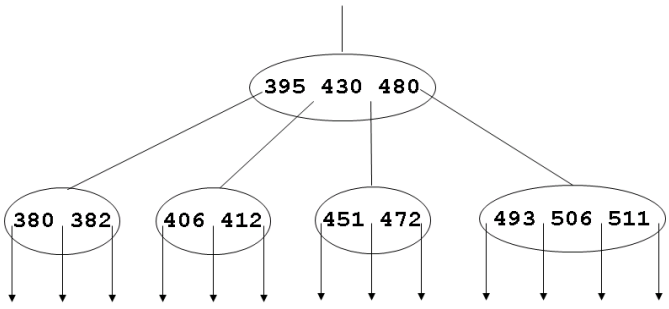
```
int d[ ]={10,11,12,13,14,15,16,17};
int max(int i, int j)
{   int n; int m=(i+j)/2;
   if (j==i) return d[i];
   else {
        if (max(i,m) > max(m+1,j)) n=max(i,m);
        else n=max(m+1,j);
        printf("%d %d %d \n", i,j,n); return n;
   }
}
void main( )
{ max(0,7); }
```

2. [3 2 8 5 7 1 4 6] is a permutation of [1 2 3 4 5 6 7 8]. Please give the two nontrivial permutation cycles in [3 2 8 5 7 1 4 6]. (6%)

3. Suppose that in a hash table, each entry can stored at most one element. When a hash collision occurs, a sequence of rehash functions h_0, h_1, h_2, h_3… are applied, where $h_i(x)=(x+3i)$ mod 10. Note that h_0 is the first hash function to be used. Please write out the hash table after all numbers have been inserted into the table according to the input order 13, 66, 43, 65, 54, 26. (10%)

4. Suppose 5,4,2,7,9,6 are inserted into an empty AVL in that order. Please draw the three trees after (a) 5,4,2; (b) 5,4,2,7,9; (c) 5,4,2,7,9,6 are inserted. (9%)

5. Please draw the tree after 490 and 495 are inserted into the following B tree of order 5. (6%)



6. Define a recursive function F as follows:

   $F(n) = 0$ if $n = 0$,

   $F(n) = n\%10$ if $(n\%10) > 0$,

$F(n) = F(n/10)$ otherwise.

(a) What is the value of $F(2)+F(3)+F(4)+\cdots+F(43)$? (6%)

(b) What is the value of $F(23)+F(24)+F(25)+\cdots+F(1234)$? (6%)

7. A red-black tree is a binary search tree in which every node is colored either red or black. Please describe the three important properties for the colors of the nodes. (9%)

8. Explain each of the following terms. (16%)
   (a) stable sorting
   (b) threaded binary tree
   (c) external path length in a binary tree
   (d) main difference between B tree and B+ tree

9. Given an input integer x, write a recursive C++ function to return y such that $x \le y$ and y-x is the minimum value, where y is the value stored in one node of a binary tree (not a binary search tree). (12%)

```
class TreeNode {
   int data;
   TreeNode *leftChild, *rightChild;
};
int minB( TreeNode *root, int x)
// Return the answer y of the binary tree pointed by "root".
// Return 99999 if the binary tree is empty or it has no answer.
{
```

Please write the body of the function.

```
} // end of minB ( )
```

10. Write a recursive C++ function to perform *Quick Sort* with nondecreasing order. You can directly call swap(x,y) to exchange the values of x and y. (12%)

```
int a[100]; // global array for storing the elements to be sorted.
            // The sorted elements are still stored in a [ ].
void swap(int &c, int &d)    // exchange the values of c and d
{ int temp;
  temp=c;    c=d;    d=temp;
}
void qsort(int left, int right) // a recursive function
// left, right: the left and right indexes of a[ ] to be sorted
{
```

Please write the body of the function.

```
} // end of qsort ( )
```

Answer:

1.

0 1 11
2 3 13
2 3 13
0 3 13
4 5 15
6 7 17
6 7 17
4 7 17
4 5 15
6 7 17
6 7 17
4 7 17
0 7 17

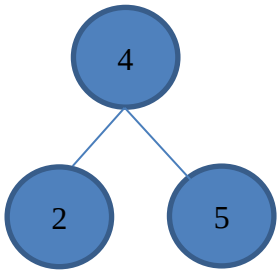
2. (1, 3, 8, 6) and (4, 5, 7)

3. hash table

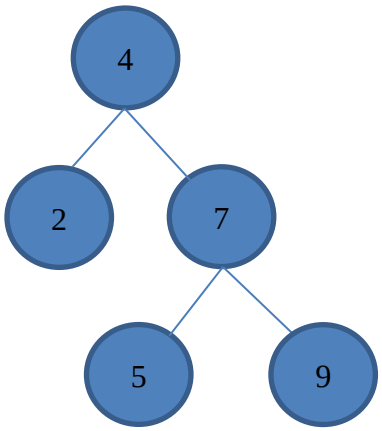
0	1	2	3	4	5	6	7	8	9
		26	13	54	65	66			43

4. AVL tree

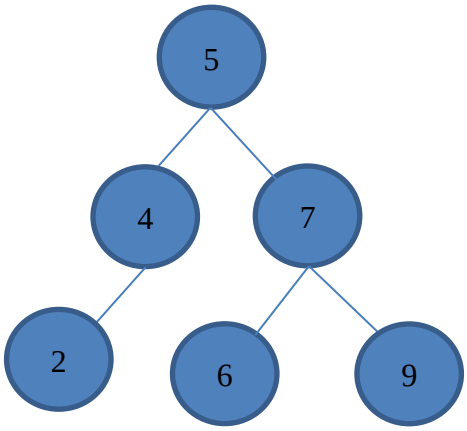
(a)



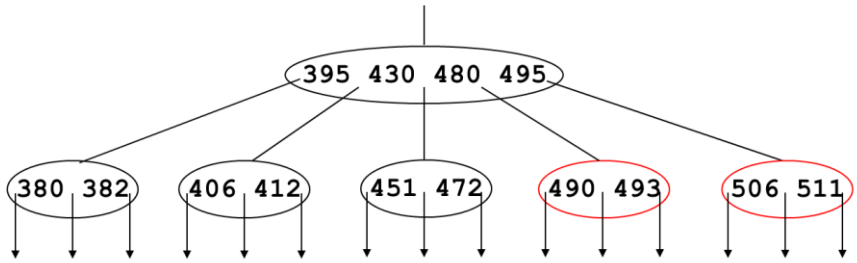
(b)



(c)



5. B tree



6. (a) 195 (b) 6044

7. red-black tree

(a) root 與 external node 都是黑色節點

(b) 從 root 到 external node 的路徑中，不會有兩個連續的紅節點

(c) 所有 root 到 external node 的路徑中，黑色節點的數量都一樣

8.

(a) stable sorting: the records with the same key have the same relative order as they have before sorting.

Example: Before sorting 6 3 7_a 9 5 7_b

After sorting 3 5 6 7_a 7_b 9

(b) threaded binary tree: A null rightChild field of node p points to the inorder successor of p. And, a null leftChild field of node p points to the inorder predecessor of p.

(c) external path length in a binary tree: The sum of all the heights of external nodes,

(d) main difference between B tree and B+ tree: In a B tree you can store both keys and data in the internal and leaf nodes, but in a B+tree you have to store the data in the leaf nodes only. Meanwhile, the elements stored in the leaf nodes of a a B+tree are maintained by a linked list.

9.

```
int minB( TreeNode *root, int x)
// Return the answer y of the binary tree pointed by “root”.
// Return 99999 if the binary tree is empty or it has no answer.
{
    int left, right, min;
    if(root == 0)
        return(99999);
    left = minB(root->leftChild, x);
    right = minB(root->rightChild, x);
    if(left < right)
```

```
        min = left;
    else
        min = right;
    if(root->data >= x && root->data < min)
        min = root->data;
    return(min);
} // end of minB ( )
```

10.

```
    void qsort(int left, int right){
        if (left < ritght) {
            int i = left, j = right + 1, pivot = a[left];
            do{
                do i++; while (a[i] < pivot);
                do j--; while (a[j] > pivot);
                if (i<j) swap(a[i], a[j]);
            } while (i < j);
            swap(a[left], a[j]);

            QuickSort(left, j–1);
            QuickSort(j+1, right);
        }
    }
```