Data Structures (COMP 2000)
Assignment 1

Available Date: Tuesday, Feb. 02, 2016
Due Date: 11.50 PM, Thursday, Feb. 18, 2016
Total Marks: 100 marks
Assessment
Coursework: 40%
Assignments (20%): A1 (7%), A2 (7%), A3 (6%)
Coursework exams (20%): CWE1 (10%), CWE2 (10%)
Final Examination: 60% (one two-hour writing exam)

Assignment Requirements

1. [20 marks] Given an unsorted array containing *n* distinct integers, denoted as *a*[0..*n* - 1]. Write complete C programs to implement the merge sort algorithms by using
a. [10 marks] the recursive algorithm. The program name is **`mergeSortR.c`**.
b. [10 marks] the iterative algorithm. The program name is **`mergeSortI.c`**.
In your programs, the given array *a* is sorted in ascending order.

2. [20 marks] Given an unsorted array containing *n* distinct integers, denoted as *a*[0..*n* - 1]. Write complete C programs to implement the quick sort algorithms.
a. [10 marks] the partition algorithm proposed by Lomuto. The program name is **`quickSortL.c`**.
b. [10 marks] the partition algorithm proposed by Hoare. The program name is **`quickSortH.c`**.
In your programs, the given array *a* is sorted in ascending order.

3. [10 marks] Given an unsorted array containing *n* distinct integers, denoted as *a*[0..*n* - 1]. Write a complete C program named **`heapSort.c`** to implement the heap sort algorithm. In your program, the given array *a* is sorted in ascending order.

4. [10 marks] Write a complete C program named **`merge.c`** to merge two sorted arrays of integers. In your program, the **`merge()`** function merges two arrays of integers *a* and *b* that are sorted in ascending order into an array of integers *c* that is also sorted in ascending order. Assume that the numbers of elements of *a* and *b* are *m* and *n*, respectively. The **`merge()`** function is called in the **`main()`** function as follows: **`c = merge(a, m, b, n)`**.

5. [10 marks] Write a complete C program named **`kSmallest.c`** to find the *k*th smallest element of an unsorted array containing *n* distinct integers, denoted as *a*[0..*n* - 1]. In your program, the **`kSmallest()`** function returns the *k*th-smallest number *a*[*k*] ($0 \le k \le n - 1$) of the given array without sorting the array.

6. [10 marks] An $n \times n$ lower-triangular matrix, denoted as *a*[0..*n* - 1][0..*n* - 1], is one in which all elements above the main diagonal are zero and the non-zero elements are all found on or below the main diagonal. Write a complete C program named **`lowerTri.c`** to transfer the elements of the matrix *a* to a one-dimensional array *b*, denoted as *b*[0..$n(n + 1)/2$ - 1] and vice versa. In your

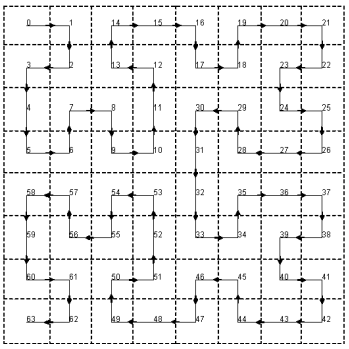
program, the **`tran2Dto1DTM()`** function stores the elements of *a* to *b* and the **`tran1Dto2DTM()`** function transfers the elements of *b* back to *a*.

7. [10 marks] An $n \times n$ matrix, denoted as $a[0..n-1][0..n-1]$ is symmetric if $a_{ij} = a_{ji}$, $0 \le i, j \le n-1$. Write a complete C program named **`symMat.c`** to transfer the elements of the symmetric matrix *a* to a one-dimensional array *b*, denoted as $b[0..n(n+1)/2-1]$ and vice versa. In your program, the **`tran2Dto1DSM()`** function stores the elements of *a* to *b* and the **`tran1Dto2DSM()`** function transfers the elements of *b* back to *a*.

8. [5 marks] Write a complete C program named **`zigzag.c`** to display the elements of a 2D array $a[0..H-1][0..W-1]$ by using the zigzag scan order as shown below.

	0	1	2	3	4
0	0	1	5	6	14
1	2	4	7	13	15
2	3	8	12	16	21
3	9	11	17	20	22
4	10	18	19	23	24

9. [5 marks] Write a complete C program named **`hilCur.c`** to display the elements of a 2D array $a[0..H-1][0..W-1]$ by using the Hilbert-curve scan order as shown below.



Submission: **carefully** submit your source program files (i.e., **`*.c`**) to Mr. Sterling Ramroach via the email: sramroach@gmail.com.

• At the top of your program, you should include the following information.

```
/* Student Full Name:
   Student ID:
   E-mail:
   Course Code:
*/
```

End of Assignment 1