BBM 201 - Data Structures - Fall 2015
Midterm 1
Date: October 26, 2015
Time: 9:30-11:20

Ad Soyad / Name:
Ögrenci No /Student ID:
Şube /Section:

Question	1	2	3	4	5	6	7	8	9	10	11	Total
Points	10	8	10	6	10	6	10	8	15	9	8	100
Grade												

1. (10 points) Decide if the following statements are true or false, circle your answer.

The cost of insertion (Push) and deletion (Pop) of an element in a stack is O(1).	True	False
The ~~complexity~~ order of growth of the worst-case running time of the 2-Sum problem for an input of size n is $n \log n$.	True	False
The ~~complexity~~ order of growth of the worst-case running time of the iterative Fibonacci algorithm for finding f_n is n.	True	False
The ~~complexity~~ order of growth of the worst-case running time of binary search for an array of size n is n^2.	True	False
$3n^2 + 10n \log n = O(n \log n)$	True	False

2. (8 points)

   (a) (4 points) Find what RecursiveFunc (given below) does **in general**.

   Finds the maximum of elements in a

   (b) (4 points) Print the output of the following program.

```
int RecursiveFunc (int array[], int index, int n)
{
     int val1, val2;
     if ( n==1 )
          return array[index];
     val1 = RecursiveFunc (array, index, n/2);
     val2 = RecursiveFunc (array, index+(n/2), n-(n/2));
     if (val1 > val2)
          return val1;
     else
          return val2;
}
...
int a[8] = 1,2,10,15,16,4,8,2;
printf( "value is % d \n", RecursiveFunc(a,0,8));
```

value is 16

3. (10 points) Below is a pseudocode of the Euclidean algorithm that calculates the greatest common divisor (gcd) of any given two integers. Write all recursive calls made to calculate `gcd(21, 1039)` by using a *box diagram.*

```
algorithm gcd(x,y)
  if y = 0
    then return(x)
  else return(gcd(y,x mod y))
```

$x = 1039$
$y = 21$

1039 | 21 = y
-84 | 49
199 − 189 = 10 x mod y

21 | 10
20 | 2
1 x mod y

10 | 1
10
0 x mod y

gcd(1039, 21)
gcd(21, 10)
gcd(10, 1)
gcd(1, 0)
return 1

4. (6 points) Write the complexity of the running time of the following code fragment using big-Oh notation. Show your work.

```
int count = N, i = 0, j = 0;
for (i = 0; i<count; i++)
     if(i%2==0) // i , even
          j++;
     else
          j--;
for (i = count; i > 0; i = i/2)
     for (j = 0; j < i; j++)
          count ++;
```

$$\left\lceil \frac{N}{2} \right\rceil + \left\lceil \frac{N}{2} \right\rceil + N + \frac{N}{2} + \frac{N}{4} + \dots + \frac{N}{2^k} =$$

$$k = \log_2 N \qquad = O(N)$$

5. (10 points) How many array accesses do the following code fragments make as a function of N?

(a) (5 points)

```
int count = 0;
for (int i = 0; i < N; i++)
     for (int j = i+1; j < N; j++)
          if (a[i] + a[j] == 0)
               count++;
```

$$2\cdot\left[(N-1)+(N-2)+\cdots+1\right]=2\binom{N}{2}=\frac{N(N-1)}{2}\cdot 2$$

(b) (5 points)

1

```
int count = 0;
for (int i = 0; i < N; i++)
     for (int j = i+1; j < N; j = 2*j)
          if (a[i] + a[j] == 0)
               count++;
```

N times

$$2\Big[\underbrace{\overset{i=0}{\log_2(N-1)}+\overset{i=1}{\log_2(N-1)}+\cdots+\overset{i=N-1}{\log_2(N-1)}}_{N\text{ times}}\Big]$$

$$\sim 2N\log_2 N$$

6. (6 points) What will be the output when doSomething(2) is called?

```
void doSomething(int value)
{
     if(0 < value && value < 10)
     {
          doSomething(value - 1);
          doSomething(value + 1);
          printf(" %d", value);
     }
}
```

dS(2)

dS(1) dS(3)

dS(2) dS(2) dS(4)

dS(1) dS(3)

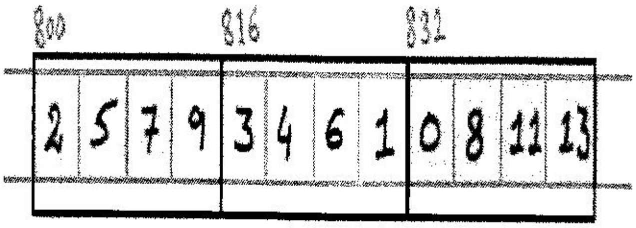
dS(2) dS(2) dS(4)

Nothing is printed.

Never Stops

7. (10 points) If C is the array shown with its address above each node, write what the following lines of a program will print in the empty column.

```
int C[3][2][2];
```



The code	*will print:*
`printf("C=%d", C);`	800
`printf("C+1=%d", C+1);`	816
`printf("*(*C+1)) =%d", *(*C+1));` *C[0][1]	808
`printf(" *(C[1]+1)=%d '', *(C[1]+1)) ;`	824
`printf(" *(C[1][1]+1)=%d '', *(C[1][1]+1));`	1

8. (8 points) Let an array be defined with `int a[6][8][5][7]`. What is the memory address of `a[3][4][2][1]` if the memory address of `a[0][0][0][0]` is x? (Assume that size of an integer is 4 bytes.)

$$x + [3 \cdot (8 \cdot 5 \cdot 7) + 4 \cdot (5 \cdot 7) + 2 \cdot 7 + 1] \cdot 4$$

9. (15 points) Let A[n][n] be an upper triangular matrix in the example given below. The elements of this triangular matrix A are stored in the one-dimensional array U as shown.

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ 0 & a_{11} & a_{12} & a_{13} \\ 0 & 0 & a_{22} & a_{23} \\ 0 & 0 & 0 & a_{33} \end{bmatrix} \begin{array}{l} 0 \\ i \\ \left.\begin{array}{l} \\ \\ \end{array}\right\} n-i-1 \end{array}$$

$U =$	a_{00}	a_{01}	a_{11}	a_{02}	a_{12}	a_{22}	a_{03}	a_{13}	a_{23}	a_{33}

(a) (~~3~~ points) What is the number of items stored in U if A has n rows and n columns?
4

$$(N+(N-1)+\cdots+1) = \frac{N\cdot(N+1)}{2} = \binom{N+1}{2}$$

(b) (~~4~~ points) What is x if $U[x]$ stores the entry $A[i][j]$? Show your calculations.
5

Answer 1: $n+(n-1)+\cdots+(n-(i-1))+(j-i) = n\cdot i - \frac{(i-1)\cdot i}{2} + j - i$

$\doteq (n-1)i - \binom{i}{2} + j$

Answer 2: $\binom{n+1}{2} - \binom{n-i+1}{2} - (n-j)$

6

(c) (~~5~~ points) Please fill in the blanks in the method readtriangularmatrix(int[], int) that reads integers from the keyboard and fills the one-dimensional array U with entries of A as shown in the example.

```
void readtriangularmatrix(int U[], int n)
{
     int i, j, k;
     if(n*(n+1)\ 2 > MAX_SIZE){
          printf("\n invalid array size \ n");
          exit(-1);
     }
     else
          for(i=0; i<=n-1; i++){
               k= ...........
               for(j=0; j<=i; j++)
                    scanf("%d", ........... );
          }
}
```

$\rightarrow \binom{n+1}{2} - \binom{n-i+1}{2}$; I

$\rightarrow$ &u[k-(n-j)] II

Or according to Answer 1:

I: $(n-1)i - \binom{i}{2}$;

II: &u[k+j]

10. (9 points) Assume a special type of matrix, named *stripe matrix*, is a matrix that has value 0 in all even-numbered columns (column 2, column 4, ...) as shown in the example below. (For simplicity, assume that all stripe matrices are N by N matrices and N is even.)

$$A = \begin{array}{c} \\ i=0 \\ i=1 \\ i=2 \\ i=3 \end{array} \begin{array}{c} \begin{array}{cccc} j=0 & j=1 & j=2 & j=3 \end{array} \\ \begin{bmatrix} 2 & 0 & 3 & 0 \\ 4 & 0 & 1 & 0 \\ 5 & 0 & -3 & 0 \\ 20 & 0 & 12 & 0 \end{bmatrix} \end{array}$$

(a) (5 points) Write a 1-dimensional array representation of the matrix above, by filling in the entries of the array U.

	U[0]	U[1]						U[7]		
U =	2	4	5	20	3	1	-3	12	X	X

(b) (4 points) What is the value of x if $U[x]$ stores a_{ij}?

$$x = n \cdot \frac{j}{2} + i - 1$$

Check: a_{30} is at $U[3] = 20$ ✓

a_{32} is at $U[4+3] = U[7] = 12$ ✓

11. (8 points) The following operations are applied to an empty queue consecutively:
`Enqueue(7)`, `Enqueue(5)`, `Enqueue(2)`, `Dequeue( )`, `Enqueue(9)`, `Enqueue(8)`, `Enqueue(11)`, `Enqueue(4)`, `Enqueue(6)`, `Enqueue(3)`, `Dequeue( )`, `Enqueue(4)`, `Enqueue(3)`

(a) (4 points) Given an array `int A[10]`, fill the entries of the array representing the queue obtained above. (Assume that the array is initially empty.) Can all operations be performed?

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
~~7~~	~~5~~	2	9	8	11	4	6	3	4

(We cannot insert 3.)

(b) (4 points) Given a *circular* array `int A[10]`, fill the entries of the circular array representing the queue obtained above. (Assume that the array is initially empty.) Can all operations be performed?