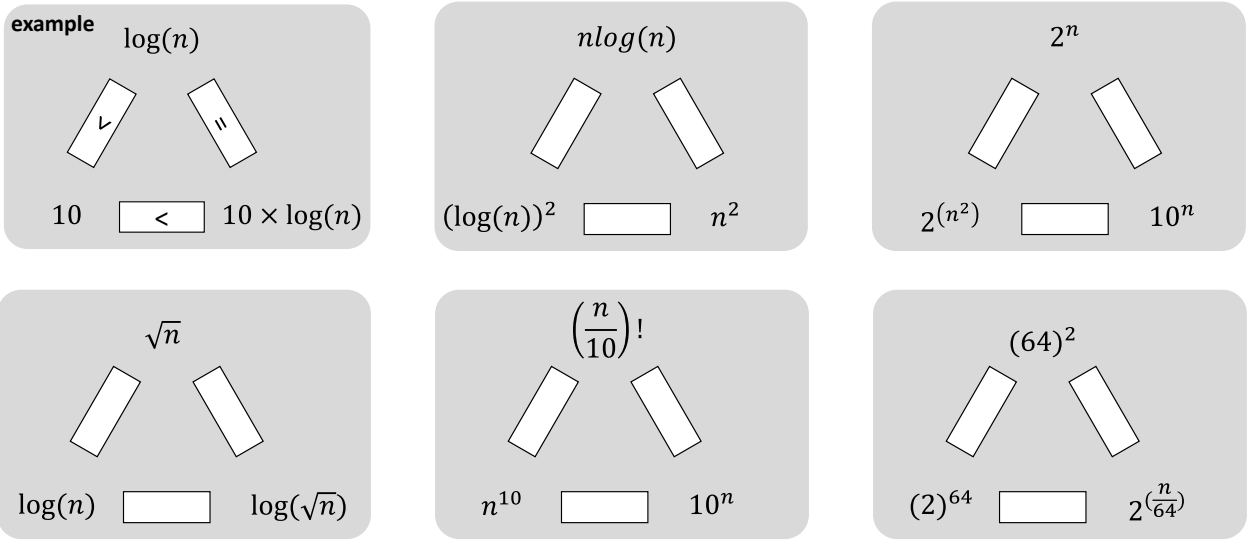
Data Structure Midterm Examination
3:30pm-5:20pm (110 minutes), Monday, April 27, 2015

- ✧ Please answer questions 1, 2, 3, and 4A on the Question Sheet. For other questions, please answer on the Answer Sheet in any order.
- ✧ There are 11 questions, each being 10 points.

1. Please find the **asymptotic order** of the following function groups:



2. Please consider the **KMP** algorithm.

A. Please analyze the **failure function** of the following pattern string.

'a'	'a'	'b'	'a'	'a'	'a'	'b'	'a'	'a'	x
-1									_____ if (x == 'a') _____ if (x == 'b') _____ otherwise

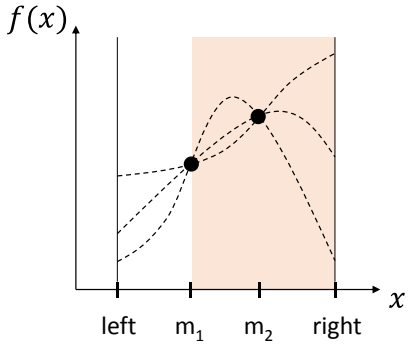
B. Please compose a pattern string that exhibits the following failure function. Please try to compose as long a string as possible and mark an 'X' to denote the position (if any) where the failure function becomes invalid.

-1	-1	0	-1	0	1	2	3	0	1
a									

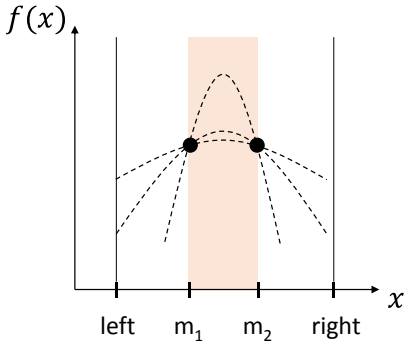
3. Please consider the infix expression A*((B-A)+3@C/D), in which '@' is a binary operator whose priority is higher than '+' and '-' but lower than '*' and '/'. Please fill in the following table that shows the procedure of infix-to-postfix using a stack.

	Stack State	**Output State**
A		
*		
(		
(		
B		
-		
A		
)		
+		
3		
@		
C		
/		
D		
)		
(End)		

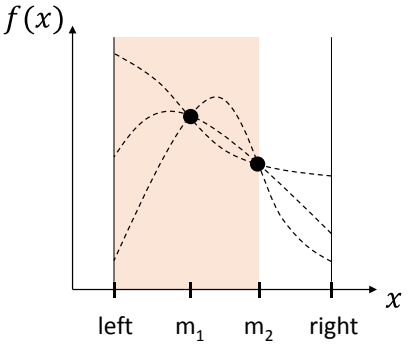
4. **Ternary Search** can be used to find the maximum of a bell-shape function. Let $f(x)$ be a bell-shape function defined on some interval [left, right] and m_1 and m_2 be two arbitrary points in the interval such that $left < m_1 < m_2 < right$. The values of $f(m_1)$ and $f(m_2)$ can exhibit three possibilities, each of which indicates a reduced interval that the maximum lies in, as depicted as follows.



if $f(m_1) < f(m_2)$
the maximun lies in [m_1, right]



if $f(m_1) == f(m_2)$
the maximun lies in [m_1, m_2]



if $f(m_1) > f(m_2)$
the maximun lies in [left, m_2]

A. Let n = (left-right+1) be the problem size. Please analyze the step count per execution (**s/e**) of the following iterative version of the **Ternary Search** algorithm:

		s/e
1:	**int** TernarySearch(**int** left, **int** right)	--
2:	{	
3:	**while**(1){	
4:	**If** (right - left <= 2)	
5:	**return** integer x that has the greatest f(x), left <= x <= right	
6:	m1 = left + (right - left)/3;	
7:	m2 = right - (right - left)/3;	
8:	**if** (f(m1) < f(m2))	
9:	left = m1;	
10:	**else if** (f(m1) > f(m2))	
11:	right = m2;	
12:	**else**{	
13:	left = m1; right = m2; }	
14:	}	
15:	}	

----- *Please answer the following questions on the Answer Sheet* -----

B. Please show a **recursive version** of the ternary search algorithm. You can directly quote the iterative version of code using line numbers.

C. Please analyze the time complexity of the recursive **Ternary Search** using the **O** notation. Try to show as tight a bound as you can.

5. Some languages allow array index to start from any arbitrary integer. Please consider a three-dimension array Z[1….20][20…70][1…15] in the row-major order with one-byte element size in this type of language. Assume Z[10][20][1] is stored at address 2000.
   A. What is the address of Z[10][30][10]?
   B. What is the array index at the location 2050?

6.
   A. Please depict a **circular, singly linked list of integers with a header node**.
   B. Please design an algorithm that can **sort the abovementioned type of list** using pseudo code.

7. You're asked to perform **postfix evaluation (note: NOT the infix-to-postfix)** using **two queues**, q1 and q2, without any stack. A queue supports **add()**, which adds an element at the rear of the queue, **remove()**, which take an element from the front of the queue, and **size()**, which reports the number of elements in the queue. Please show your algorithm using pseudo code.

8. Short answer questions / explanation of terminologies
   A. What issue does **C++ template** aim to address?
   B. What are the pros and cons of **data encapsulation?**
   C. Is it possible that an $\mathbf{O}(2^n)$ (i.e., exponential-time) algorithm outperforms an $\mathbf{O}(n)$ (i.e., linear-time) algorithm in terms of speed? How can this occur?

9. Please proof that
   if $\mathrm{F(n)} = \mathbf{O}(\mathrm{G(n)})$, then $(\mathrm{F(n)} + \mathrm{G(n)}) = \mathbf{O}(\mathrm{G(n)})$.

10. Please analyze the **worst-case time complexity** of the following procedure with **brief explanation**. Please find as tight a bound as you can.

```
1:  // in[][] is an N-by-M input array
2:  int a=1, b=1;
3:  while ( a<N && b<M ) {
4:      if( in[a][b] == true )   { a++;   b = b*a;     }
5:      else if ( b == 0 )       { a++;                }
6:      else                     { b = b/2;            }
7:  }
```

11. We want to design a **circular queue** class that is implemented in terms of a **16-element integer array** and can store up to **15 integers**. Please show **add()**, **remove()**, and **size()** operations using **pseudo code**. These operations should all be of $\mathbf{O}(1)$ time complexity.