Please write your answers to Parts A and B in the spaces below.

<table>
<thead>
<tr>
<th>Part A</th>
<th>Part B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. T</td>
<td>11. D</td>
</tr>
<tr>
<td>2. T</td>
<td>12. C</td>
</tr>
<tr>
<td>3. F</td>
<td>13. C</td>
</tr>
<tr>
<td>5. T</td>
<td>15. D</td>
</tr>
<tr>
<td>6. F</td>
<td>16. B</td>
</tr>
<tr>
<td>7. T</td>
<td>17. B</td>
</tr>
<tr>
<td>8. F</td>
<td>18. C</td>
</tr>
<tr>
<td>9. T</td>
<td>19. A</td>
</tr>
<tr>
<td>10. T</td>
<td>20. A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scoreboard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part A</td>
</tr>
<tr>
<td>Part B</td>
</tr>
<tr>
<td>Part C</td>
</tr>
<tr>
<td>Part D</td>
</tr>
<tr>
<td>Part E</td>
</tr>
<tr>
<td>Part F</td>
</tr>
<tr>
<td>Part G</td>
</tr>
<tr>
<td>Part H</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>
Part A: True or False

Write your answers on the front page (10 pts).

1. Merge sort is a stable sort.

2. The function \( F(n) = n + \lg n \) is \( O(n \lg n) \).

3. The function \( G(n) = n + \lg n \) is \( \Omega(n \lg n) \).

4. For a Max-heap, the operation of getting the number of external nodes can be performed in constant time.

5. A good hash table has a search time of \( O(1) \).

6. The greedy technique can always yield the optimal solution.

7. The worst-case time complexity for searching for an element in an unsorted array is \( O(n) \).

8. An \( O(n^3) \) function always grows faster than an \( O(n) \) function.

9. Depth first search can be used to detect cycles in graphs.

10. In a binary search tree, for a node with two children, the value of the left child must be lower than the value of the right child.
Part B: Multiple Choice

Write your answers on the front page (10 pts).

11. Which of the following is a valid bound for $F(n) = \lg n + n^{10}$?

(a) $O(1)$
(b) $O(\lg n)$
(c) $O(n^9)$
(d) $\Omega(\lg n)$

12. Which sequence will result in a worst running time for quick sort? (Assume we always select the first element as pivot.)

(a) 9 3 4 6 8 2 1
(b) 1 3 6 9 2 4 8
(c) 9 8 6 4 3 2 1
(d) 2 3 1 6 8 9 4

13. Which of the following is not an example of ”Divide and Conquer”?

(a) Merge sort
(b) Quick sort
(c) Bubble sort
(d) None of the above

14. Choose the wrong statement regarding the collision probability of a hash table of size $n$.

(a) Increasing the load factor ($\alpha$) can increase the collision probability.
(b) Open addressing does not affect the collision probability.
(c) The collision probability of a universal hash function is not greater than $1/n$.
(d) The bigger the size $n$, the lower the collision probability.
15. Height of the quick-sort tree in the worst case is:

(a) $O(n \log n)$
(b) $O(n^2)$
(c) $O(\log(n - 1))$
(d) $O(n - 1)$

16. Which operation is NOT supported by the min-priority queue abstract data type?

(a) minKey()
(b) remove(key)
(c) insertItem(key, value)
(d) removeMin()

17. With an empty stack, choose the minimum capacity of the stack such that the following operations are performed without a stack overflow error:

push(1), push(2), pop(), push(3), pop(),
push(4), push(5), pop(), pop(), push(6).

(a) 2
(b) 3
(c) 4
(d) 5

18. Which of the following is a min-heap?

(a) 614257
(b) 154672
(c) 152674
(d) 756241
19. What is the time complexity of the following code segment?
for (i = 0; i < n; i++)
    for (j = m; j > 0; j = j-1)
        for (k = n; k > 0; k /= 2)
            l++;
(a) $\Theta(nm \lg n)$
(b) $\Theta(n \lg n)$
(c) $\Theta(\lg n^2)$
(d) $\Theta(nm - \lg n)$

20. What is the content of the following array after two rounds of radix sort?

< 20, 120, 21, 11, 201, 2, 102, 10 >

(a) < 201, 2, 102, 10, 11, 20, 120, 21 >
(b) < 2, 10, 11, 20, 21, 102, 120, 201 >
(c) < 2, 201, 102, 10, 11, 20, 120, 21 >
(d) < 10, 102, 11, 120, 2, 20, 201, 21 >
Part C

Consider the following functions:

\[
\begin{align*}
p(n) &= 1 \\
q(n) &= n^{50} \\
r(n) &= 50n \\
s(n) &= n \log n \\
t(n) &= n^n \\
u(n) &= n + \log n \\
v(n) &= 2^n \\
w(n) &= \log n
\end{align*}
\]

i) Which two functions are \( \Theta \) of one another (i.e. they have the same order of growth) (5 pts)?

[A]: \( r(n) \) and \( u(n) \)

ii) Rank the functions in ascending order of growth with the two functions from the previous question at the same rank (5 pts).

[A]: \( p(n) < w(n) < r(n) = u(n) < s(n) < q(n) < v(n) < t(n) \)
Part D

Consider the following binary search tree (BST):

```
      10
     /   \
     4    21
    /     |
   2      25
   |      / |
  8    9    26
```

i) Give the post-order traversal of the BST (5 pts).
[A]: 2, 8, 9, 4, 26, 25, 21, 10

ii) Draw the resulting tree after inserting 22 into the BST (5 pts).
[A]:

```
      10
     /   \
     4    21
    /     |
   2      25
   |      / |
  8    9    22
   |
   26
```

iii) Continuing from the previous question, draw the resulting tree after removing 21 from the BST (5 pts).
[A]:

```
      10
     /   \
     4    25
    /     |
   2      22
   |
   9
```

Part E

Assume we have a hash function: \( h(x) = x \mod 13 \) and a hash table of size 13. Linear probing is used to handle collisions.

i) Illustrate the final hash table after performing the following operations (5 pts):
insert 0, insert 1, insert 5, insert 7, insert 13,
insert 17, insert 19, insert 15, remove 1.

\[
\begin{array}{cccccccccccc}
0 & available & 13 & 15 & 17 & 5 & 19 & 7 & 8 & 9 & 10 & 11 & 12 \\
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
\end{array}
\]

ii) Describe a sequence of insertions that would lead \( \text{search}(x) \) to run in worst-case time (5 pts).

[A]: Insert 0, 13, 26, 39, 52, 65, 78, 91, 104, 117, 130, 143, 156 in turn. Their hashcodes are all 0. The hash table will end up full and \( \text{search}(156) \) will require \( O(n) \) time.

iii) How can universal hashing prevent the situation in the previous question? (5 pts)

[A]: Universal hashing randomly chooses hash functions to reduce the probability of a collision between two different keys to no more than \( 1/m \) chance of just choosing two slots randomly and independently.

Part F

i) What is the time complexity of searching an element in a skip-list? (5 pts)

[A]: \( O(\log(n)) \)

ii) Does a skip-list have the same \textit{space}-complexity in terms of \textit{big} O notation as a linked-list? Why or why not? (5 pts)

[A]: Yes. Please check the last slide of Lecture5.pdf for why.
Part G

Use quicksort to sort this sequence: < 2, 8, 12, 7, 1, 13, 4, 5 >. Assume we always choose the first element as a pivot. Illustrate your results after each pass. (15 pts)

[A]:

```plaintext
<2, 8, 12, 7, 1, 13, 4, 5>
<1>   <8, 12, 7, 13, 4, 5>
<5, 4, 7>   <12, 13>
<4>   <7>   <13>
```
Part H

Consider this Java snippet for finding the number of nodes in a binary tree:

class Node {
    Node left;
    Node right;
}

int nodeNumber( Node node ) {
    // FILL THIS IN
}

In Java syntax, give your implementation (15 pts).

[A]:

int nodeNumber( Node node ) {
    if( null == node ) {
        return 0;
    }
    return nodeNumber( node.left ) + nodeNumber( node.right ) + 1;
}