This exam is closed book, notes, computer and cell phone. 
Show partial solutions to get partial credit. 
Make sure you have answered all parts of a question. 
If your solutions are not written legibly, you won’t get full credit. 
Clarity and succinctness will be rewarded.

Question 1:___________(out of 10) 
Question 2:___________(out of 10) 
Question 3:___________(out of 10) 
Question 4:___________(out of 10) 
Question 5:___________(out of 10) 
Question 6:___________(out of 10) 
Question 7:___________(out of 10) 
Question 8:___________(out of 10) 
Question 9:___________(out of 10) 
Question 10:__________(out of 10) 

Total:_____________(out of 100)
1. SHORT QUESTIONS:

   (a) Which tree traversal does BFS correspond do?

   (b) Which graph traversal does backtracking correspond do?

   (c) Give one application of BFS?

   (d) What is the running time of BFS and DFS
       i.t.o. the number of vertices $n$ and the number of edges $e$ in the graph
       (assume the graph is given in the adjacency list representation)

   (e) What is the time to build a min-heap with $n$ arbitrary keys?
2. You are to implement the Delete operation for the array based implementation of a min heap. The keys of the heap are stored in an almost complete binary tree which is stored in an array from index one to last.

Tree figure:

```
       5
      / \
     7   9
    /  / \
   20 21 60 70
  /   / \
90   22 51
```

Array representation:  
<table>
<thead>
<tr>
<th>index i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>A[i]</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>20</td>
<td>21</td>
<td>60</td>
<td>70</td>
<td>90</td>
<td>22</td>
<td>51</td>
</tr>
</tbody>
</table>

Delete(A, last, index) 
(* deletes the key A[index] from the heap A[1..last] *)

- What is the tree and the array after you delete the node containing the key 7.

- Give high level pseudo code for your Delete routine.

- What is the running time of your algorithm and why?
3. The following recurrence for computing the Fibonacci numbers immediately leads to a recursive program:

\[
F(0) = 0 \text{ and } F(1) = 1 \\
\forall n \geq 2 : F(n) = F(n-1) + F(n-2)
\]

- Give the pseudo code of a short recursive program for computing the \(n\)th Fibonacci number

- What is the problem with this program?
  Roughly how many calls does it incur when called with argument \(n\)?

- Give a Dynamic Programming algorithm that computes the \(n\)th Fibonacci number

- Reason that the time for computing the \(n\)th number is \(O(n)\)
  (assuming that adding two numbers is \(O(1)\))
4. • At a high level, describe the ”multiplication method” for creating a hash function.

• When open addressing is used, why is double hashing better than linear and quadratic probing?
5. Show the red-black trees that result after successively inserting the keys

52 48 33 10 20 0

into an initially empty red-black tree.
6. Given the following two BHs, what is the BH produced by the Union operation?
   Hint: What binary number does each binary heap correspond to?

Two binomial heaps, A and B

- Consider the operation Delete(H,x) for Binomial Heaps
  where H is a binomial heap, x a pointer to a node in the heap, and the goal is to delete
  x from H
  Express this operation i.t.o. two other BH operations.
  What is the running time of your method?
7. Consider the Disjoint-set Forests data structure. For every node \( x \), \( x.p \) points to its parent. Roots point to themselves.

Find-Set(\( x \)) returns the root of the tree (representative of the set) to which node \( x \) belongs to.

- Apply the heuristic of “path compression” as you search for the root starting from node \( x \) in the following example forest:

![Union Find Forest Diagram](image)

- Write pseudo code for the Find-Set(\( x \)) procedure (Give either a recursive or an iterative version).

- What is the name of the other heuristic that makes the Union Find data structure based on disjoint set forests efficient?

- What is the worst case running time for \( m \) operations from the set \{Make-set, Find-set, Union\} when the number of elements is \( n \)? Why is this running time “essentially linear” in \( m \).
8. Design a data structure which achieves the following:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Time Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Insert(key)</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>Search(key)</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>Delete(node)</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>Min</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Max</td>
<td>$O(1)$</td>
</tr>
</tbody>
</table>

- Can you use implementation based on both a min and a max heap? Why not?

- Use a modified RBT to implement all the operations in the required time. Sketch the modification to each operation that is required and reason that the required running time is achieved.
9. The length of the longest common subsequence between two strings \( x_1, x_2, \ldots, x_m \) and \( y_1, y_2, \ldots, y_n \) of length \( m \) and \( n \), respectively, can be computed via dynamic programming by building a table \( T[0..m, 0..n] \). In the below example, \( m = 7 \) and \( n = 6 \).

\[
\begin{array}{ccccccc}
 & & & & & & \\
\mid i \mid & x_i & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
\mid j \mid & y_j & B & D & C & A & B & A \\
0 & x_i & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & A & 0 & 0 & 0 & 0 & 1 & 1 \\
2 & B & 0 & 1 & 1 & 1 & 1 & 2 & 2 \\
3 & C & 0 & 1 & 1 & 2 & 2 & 2 & 2 \\
4 & B & 0 & 1 & 1 & 2 & 2 & 3 & 3 \\
5 & D & 0 & 1 & 2 & 2 & 2 & 3 & 3 \\
6 & A & 0 & 1 & 2 & 2 & 3 & 3 & 4 \\
7 & B & 0 & 1 & 2 & 2 & 3 & 4 & 4 \\
\end{array}
\]

Table entry \( T[i, j] \) is the length of the longest common subsequence between the partial strings \( x_1, x_2, \ldots, x_i \) and \( y_1, y_2, \ldots, y_j \).

The recurrence for computing the table is the following:
If \( x_i = y_j \) then \( T[i, j] = T[i - 1, j - 1] + 1 \)
else \( T[i, j] = \max(T[i - 1, j], T[i, j - 1]) \).

Assume you have just computed the table, and now you are to retrieve a longest common subsequence of length \( T[m, n] \).

Give a high level procedure for how to do this.

Your running time must be \( O(n + m) \).
Reason your running time.

Hint: start in the lower left corner and decide what to do based on whether \( x_i = y_j \).
10. Show by induction that

\[ \forall n \geq 1 : \sum_{i=1}^{n} x^i = \frac{x^{n+1} - x}{x - 1}. \]

- What is the base step?

- What is the induction hypothesis?

- What is the induction step?