1. Data types.

(a) A data type is a set of values and operations on those values.

<table>
<thead>
<tr>
<th>Java expression</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>r.length()</td>
<td>int</td>
<td>5</td>
</tr>
<tr>
<td>r.charAt(r.length())</td>
<td>run-time exception</td>
<td></td>
</tr>
<tr>
<td>(t == u)</td>
<td>boolean</td>
<td>false</td>
</tr>
<tr>
<td>u.equals(t)</td>
<td>boolean</td>
<td>true</td>
</tr>
<tr>
<td>r.substring(0, r.length())</td>
<td>String</td>
<td>&quot;Hello&quot;</td>
</tr>
</tbody>
</table>

2. Scientific computation.

(a) (ii)
(b) 1/2, 3/4, 1, 123

3. Linked structures.

I and III only

4. Data type design.

(a) Y declaring instance variables to be private
N declaring instance variables to be immutable
   no such access modifier in Java
Y declaring instance variables to be final
Y defensively copying instance variables
N overloading instance methods
   a feature of Java methods, but not related to immutability

(b) double[] Tour Stack<String> GuitarString
5. Analysis of algorithms.
   
   (a) \( N^2 \)
   
   (b) \( 240 = 15 \times 4^2 \)
   
   (c) \( 8N^2 \)


   ![Symbol Table Diagram]

7. Regular expressions.
   
   ii (i matches \( aA \); iii matches \( Aa \); iv doesn’t match \( a \))

8. Theory of computation.

   (a) B There exists a mathematical function that can be computed in Java, but cannot be computed on a Turing machine.

   D There exists a mathematical function that can be computed in polynomial time on a quantum computer, but cannot be computed in polynomial time on a Turing machine. *Assume that quantum computers can be built.*

   B There exists a mathematical function that can be computed in polynomial time in Java, but cannot be computed in polynomial time on a Turing machine.

   A There exists a Universal Turing machine that can simulate the behavior of any other Turing machine.

   A. known to be true
   
   B. known to be false
   
   C. if true would falsify the Church-Turing thesis
   
   D. if true would falsify the extended Church-Turing thesis
   
   E. if true would prove the Church-Turing thesis
(b) D Not all search problems can be solved in polynomial time.

A There exists a search problem that can be solved in polynomial time.

C Both FACTOR and 3-Sat can be solved in polynomial time.

B Exactly one of 3-Sat and Tsp can be solved in polynomial time.


(a)

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(b) $X'Y'Z' + X'YZ' + XY'Z + XYZ$

(c) $2^{(N+1)/2}$

There is one entry in the truth table (and an $N$-input AND gate) for each $N$-bit palindrome. For odd $N$, the first $(N + 1)/2$ bits can be 0 or 1; the last $(N - 1)/2$ bits equal the reverse of the first $(N - 1)/2$ bits.