1. Properties of reference types.
   J E A C G B

2. Object-oriented programming.
   C E I B H F A J K

3. Linked structures.
   H A I/A D

4. Sorting and searching.
   (a) 17–22, 22–33, 10–22, 10–33
   (b) 44–88, 77–88

5. Symbol tables.
   (a) 88 33 11 55 44 77 99
   (b) $n \log n$
      A worst-case input is if all $n$ integers are distinct. In this case, the while loop will
      construct a BST containing $n$ key–value pairs. So, we expect symbol-table operation to
      take $\log n$ time each. The while loop calls get() and contains() $n$ times each. The
      double nested for loop calls get() exactly $n$ times.
   (c) Sorts.
      It reads integers from standard input and prints them in ascending order to standard
      output.

6. Regular expressions and DFAs.
   (a) (a|b)a*(a|b) or .*a.
   (b) start state: 0
      accept states: 2 and 3
      a transitions: 2 $\rightarrow$ 2 and 3 $\rightarrow$ 1
      b transitions: 0 $\rightarrow$ 0 and 1 $\rightarrow$ 3

(a) C A B G

(b) D E A F

8. Circuits.

<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>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</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>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ f = xyz + xy'z' + x'yz' + x'y'z \]

```java
public static boolean f(boolean x, boolean y, boolean z) {
    if (x && y) return !z;
    if (x || y) return z;
    return !z;
}
```