Solution to the DM sample test

1. (a) Use the truth table:

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>¬q</th>
<th>p ∧ ¬q</th>
<th>¬ (p ∨ ¬q)</th>
<th>¬ p</th>
<th>q ∧ ¬p</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

Since the fifth and seventh columns agree, we conclude that ¬(p ∨ ¬q) and q ∧ ¬p are equivalent.

(b) By De Morgan’s laws ¬(p ∨ ¬q) and ¬p ∧ ¬q are equivalent. By the double negation law, this is equivalent to ¬p ∧ q, which is equivalent to q ∧ ¬p by the commutative law. We conclude that ¬(p ∨ ¬q) and q ∧ ¬p are equivalent.

2. This is false. For a counterexample take A={1,2}, B={1}, and C={2}. We have A-(B ∩ C)={1,2}-φ={1,2}, while (A-B) ∩ (A-C)={2} ∩ {1}=φ

3. If f(n)=f(m), then 2n+1=2m+1. It follows that n=m. Hence f is one-to-one. Since f(n)=2n+1 is odd for every integer n, it follows that f(n) is not onto; for example, 2 is not in its range.

4. We have f(n)=3n^2+8n+7≤3n^2+8n^2+7n^2=18n^2 whenever n≥1. It follows that f(n) is O(n^2), since we can take C=18 and k=1 in the definition.

5. We have x+2 is O(x) since x+2≤2x for all x≥2; log(x^2+1) is O(logx) since log(x^2+1)≤log(2x^2)=log2+2log x≤3log x whenever x≥2; and similarly log(x^2+1) is O(logx). It follows that (x+2)log(x^2+1) is O(xlogx) and consequently f(x) is O(xlogx).

6. (a) We first compare the first and second integers in the sequence a_1, a_2, ..., a_n, setting the value of the variable firstmax equal to the larger, and the value of the variable secondmax equal to the smaller. For each successive integer a_i in the sequence, i=3,4,...,n, we first compare it to firstmax. If a_i > firstmax, the we make the assignments secondmax := firstmax and firstmax := a_i. Otherwise, we compare a_i to secondmax, and if a_i > secondmax, then we make the assignment secondmax := a_i. At the end of this procedure the value of secondmax will be the second largest integer in the sequence.

(b). We do one comparison at the beginning of the algorithm to determine whether a_1 or a_2 is larger. Then for each successive term, for i=3,4,..., n, we carry
out at most two comparisons. Hence the largest number of comparisons used is 2(n-2)+1=2n-3, ignoring bookkeeping. This is O(n).

7. Suppose that m is even and n is odd. Then there are integers j and k such that m=2j and n=2k+1. If follows that m+n=2j+2(2k+1)=2(j+k)+1=2l+1, where l=j+k. Hence m+n is odd.

8. The basis case holds since 3^7=2187<5040=7!. Now suppose that 3^n<n! where n is a positive integer greater than or equal to 7. Using the inductive hypothesis we see that 3^{n+1}=3*3^n<3*n!<n+1)*n!=(n+1)!. This completes the proof.

9. The error is that no basis case has been done.

10. By the product rule for counting there are 26*10*10*10=26,000 different labels for lockers.

11. (a) There are 8^3=512 functions.

   (b) There are 8*7*6=336 one-to-one functions.

   (c) There are obviously no onto functions.

12. We have f(n)=6f(n-1) whenever n is a positive integer where f(n) is the number of fruit flies after n weeks, with f(0)=12.

13. (a) Let a_n be the number of ways to climb n stairs. Suppose that n≥4. Then a_n = a_{n-2} + a_{n-3}, since n stairs can be climbed by going up n-2 stairs followed by a step of 2 stairs or by going up n-3 stairs followed by step of 3 stairs.

   (b) We see that a_1 =0, a_2 =1, and a_3 =1.

   (c). Note that a_4 = a_2 + a_1 =1+0=1, a_5 = a_3 + a_2 =1+1=2, a_6 = a_4 + a_3 =1+1=2, a_7 = a_5 + a_4 =2+1=3, and a_8 = a_6 + a_5 =2+2=4.

14. (a) R_2 is reflexive since it contains (1,1), and (2,2) and (3,3). The relations R_1, R_3, and R_4 are not reflexive since they do not contain all three of these ordered pairs.

   (b) R_1 and R_3 are symmetric since they contains (i,j) whenever they contain (j,i). To check this for R_1 requires only that we note that both (1,3) and (3,1) are in the relation and to check this for R_3 requires only that we note that both (1,2) and (2,1) are in the relation. R_2 and R_4 are not symmetric since each contains one of (1,3) and (3,1), but not the other.

   (c) R_2 and R_4 are antisymmetric since neither contains ordered pair (i,j) and (j,i) where i≠j. To check this for R_2 requires only that we check that (1,3) is not in R_2, since (3,1)
is the only ordered pair in the relation with different first and second elements; to check this for $R_4$ requires only that we check that neither (3,1) nor (3,2) is in the relation. We see that $R_1$ is not antisymmetric since both (1,3) and (3,1) are in $R_1$. We see that $R_3$ is not antisymmetric since both (1,2) and (2,1) are in $R_3$.

(d) $R_2$ and $R_4$ are transitive. This is easily verified since neither has pairs $(a,b)$ and $(b,c)$ with $a \neq b$ and $b \neq c$. $R_1$ is not transitive since (3,1) and (1,3) belong to $R_1$ but (3,3) is not in $R_1$. $R_3$ is not transitive since (1,2) and (2,1) belong to $R_3$ but (1,1) does not belong to $R_3$.

15. (a) Let $x$ be a bit string. Then $(x,x) \in R$ since $x$ has the same number of 0s as itself. Hence $R$ is reflexive. Now suppose that $(x,y) \in R$. Then $x$ and $y$ have the same number of 0s. Consequently $y$ and $x$ have the same number of 0s. It follows that $(y,x) \in R$. Next, suppose that $(x,y) \in R$ and $(y,z) \in R$. Then $x$ and $y$ contain the same number of 0s and $y$ and $z$ contain the same number of 0s. It follows that $x$ and $z$ contain the same number of 0s. Hence $(x,z) \in R$. We conclude that $R$ is transitive.

(b) The equivalence class of 1 is the set of all bit strings that contain no 0s; explicitly, $[1]_R = \{\lambda, 1, 11, 111, 1111, \ldots\}$. The equivalence class of 00 is the set of all bit strings that contain exact two 0s, that is $[00]_R = \{00, 100, 010, 001, 1100, 1010, 1001, 0101, 0011, \ldots\}$. The equivalence class of 101 is the set of all bit strings that contain exact one 0s, that is $[101]_R = \{0, 10, 01, 110, 101, 011, 1110, 1101, 1011, 0111, \ldots\}$.

16. The following finite-state machine models the vending machine.

Key: $b =$ button, $n =$ nothing, $oj =$ orange juice

17. The output produced is 10000.
18. The following finite-state machine produces a 1 if and only if the last three input bits read all 0s.

19. The following finite-state automaton recognizes the set represented by the regular expression $10^*$. 

![Diagram of a finite-state machine](image1)

![Diagram of a finite-state automaton](image2)