Section 5.1

4.

$12 * 2 * 3$

8.

Here's one way to count: Take the number of all 3-letter initials, 26^3, and substruct the number of 3-letter initials that have one repetition, which is $3 * 26 * 25$ (because there's three possibilities for where the non-doubled letter is, 26 choices for the doubled letter, and 25 choices for the non-doubled letter), and 3-letter initials, which is 26. So the answer is $26^3 - (3 * 26 * 25 + 26)$.

20.

(a) $\lfloor 999/7 \rfloor$.[1]

(b) $\lfloor 999/7 \rfloor - \lfloor 999/77 \rfloor$, because you have to substruct numbers that are divisible by *both* 7 and 11, which is the same set as the set of numbers that are divisble by 77.

(c) $\lfloor 999/77 \rfloor$, see above.

(d) $\lfloor 999/7 \rfloor + \lfloor 999/11 \rfloor - \lfloor 999/77 \rfloor$, using the inclusion-exclusion principle and using part (c).

(e) $(\lfloor 999/7 \rfloor - \lfloor 999/77 \rfloor) + (\lfloor 999/11 \rfloor - \lfloor 999/77 \rfloor) = \lfloor 999/7 \rfloor + \lfloor 999/11 \rfloor - 2\lfloor 999/77 \rfloor$. It's the same argument as in part (b), but used twice, first for all numbers divisible by 7 and not 11, and then for all those that are divisible by 11 but not by 7.

(f) $999 - (\lfloor 999/7 \rfloor + \lfloor 999/11 \rfloor - \lfloor 999/77 \rfloor)$, by part (d).

(g) $9 + 9 * 9 + 9 * 9 * 8$, summing the counts of numbers with no repetitions among digits which have, respectively, exactly one digit, exactly two digits, and exactly three digits. Note that in each case the first digit has to be non-zero, so there is 9 choices, and the other digits can be any digit among all 10 digits except of the previously taken digits, so there are 9 choices of the second digit, 8 choices of the third digit, etc.

(h) This one was tricky! As I went over it in the class, there's many incorrect paths one can fall into, but here's a way that works: Let o_i be the count of i-digit numbers that satisfy this constraint, i.e. they are even and all their digits are different. The answer is $o_1 + o_2 + o_3$. We have that $o_1 = |\{2, 4, 6, 8\} = 4$. We also have $o_2 = 4 * 8 + 1 * 9$, because if the number has two digits then if its last digit is non-zero (4 choices) then its first digit has 8 choices (any digit except zero and the one that's equal to the last digit), and if its last digit is zero (1 choice) then its first digit has 9 choices (any digit except zero). Finally we have $o_3 = 4 * 8 * 8 + 1 * 9 * 8$ because if the number has three digits then if its last digit is non-zero (4 choices) then its first digit has 8 choices (any digit except zero and the one that's chosen as the last digit) and its middle digit has also 8 choices (any digit except of the two that are chosen as first and last digit), and if its last digit is zero (1 choice) then its first digit has 9 choices (any digit except zero) and its middle digit has 8 choices (same as above). So the final answer is $4 + (4 * 8 + 9) + (4 * 8 * 8 + 9 * 8)$.[2]

30.

(a) 26^8

(b) $26 * 25 * 24 * 23 * 22 * 21 * 20 * 19 = 26!/18!$, which is also $P(26, 8)$, the number of 8-permutations of 26, but this notation was introduced in Monday's lecture.

(c) 26^7, because once the first letter is fixed there's only 7 letters to choose, each from 26 possibilities.

(d) $25!/18! = P(25, 7)$, because once the first letter is fixed one has to choose 7 different letters among the remaining 25.

(e) 26^6, because two letters are fixed so there's 6 letters to choose, each from 26 possibilities.

(f) same as (e), 26^6: it does not matter which letters you fix and how as long as you fix the same number of them.

---

[1] It's actually not $\lfloor 1000/7 \rfloor$, which is a term I used when talking about this problem in class and in the explanations to the homework posted on-line. $\lfloor 1000/7 \rfloor$ would be the right answer if the question was about numbers which are less *or equal* to 1000, while here we have numbers which are strictly less than 1000. But if you used 1000 instead of 999 everywhere you'll get full credit. Getting the boundary correctly as 999 instead of 1000 was not the main point of this exercise...

[2] If you sum up the answer to (g) you get 738 and to (h) you get 364, so it's a bit more than a half, but this makes sense: count the number of odd numbers that have each digit different and you'll see that there's slightly fewer of them, and the two numbers sum up to 738.

(g) 26^4, because now four letters are fixed so there's four left.

(h) since we have "or" rule here a good way to count is using the inclusion-exclusion principle, i.e. add all strings that start with BO (26^6 of these) and all strings that end with BO (26^6 of these) and substruct all strings that both start and end with BO (26^4 of these), so the answer is $2 * 26^6 - 26^4$. (You have to substract the intersection because otherwise you'd be counting the objects of the last category twice, first time among those in the first category and the second time among those in the second category.)

34.

2^n, because each function is an assignment between set $D_n = \{1, ..., n\}$ and $B = \{0, 1\}$, and for any argument $x \in \{1, ..., n\}$ there's 2 ways to assign its value in the set $\{0, 1\}$, so the total number of assignemnts is $2 * 2 * ... * 2$ (n times), which is 2^n.

This is also the number of n-bit strings, because each function from D_n to B can be uniquely represented by an n-bit string: the i-th bit tell you the value of the function on argument i.

38.

Easier to count all subsets (2^{100}) and then substract the number of subsets that *don't* have more than one element, which are the empty subsets, of which there's only one, and one-element subsets, of which there's 100. So the answer is $2^{100} - 101$.

60.

The claim is that if $S = S_1 \times ... \times S_n$ then $|S| = |S_1| * ... * |S_n|$. The base case $n = 1$ is true by definition. In the inductive case, if $|S| = |S_1| * ... * |S_k|$ for $S = S_1 \times ... \times S_k$ for any sets $S_1, ..., S_k$, then $S' = S_1 \times ... \times S_k \times S_{k+1}$ can also be represented as $S' = S \times S_k$, and therefore it has $|S| * |S_k|$ elements by the product rule for two tasks, and by the inductive assumption this is equal to $(|S_1| * ... * |S_k|) * |S_{k+1}|$, and this proves the inductive case.

Section 5.2

4.

(a) five, because $n = 5$ is the first integer s.t. $\lceil n/2 \rceil \geq 3$. Here balls are balls and boxes are two color types. The question is how many balls do you need for some box (i.e. one of the two colors) to have at least 3 balls. By Generalized Pigeonhole Principle (GPP), this is true for n s.t. $\lceil n/2 \rceil \geq 3$.

(b) 13: this is not a PP problem, this is just simple counting. With 12 she can pick 10 red balls and 2 blue ones. With 13 she has to pick at least 3 blue ones because there's no more red balls than 10.

6.

Here the balls are the integers, so $n = d + 1$, and the boxes are the possible values of remainders modulo d, so there are $k = d$ boxes. Integer x is placed in box number ($x \bmod d$). Since $n > k$, by PP there's one box with at least two balls, i.e. two integers with same remainder modulo d.

8.

Here balls are elements of the domain S, so $n = |S|$, and boxes are elements of the range T, so $k = |T|$. Since $n > k$, by PP, there must exist two elements in S which get mapped to same value in T.

16.

Note that whether there exists a pair which sums up to 16 depends on whether or not elements i and $16 - i$ are together in the chosen subset. So define the boxes as follows: $B_1 = \{1, 15\}$, $B_2 = \{3, 13\}$, $B_3 = \{5, 11\}$, $B_4 = \{7, 9\}$. If you have two balls in any one of these four boxes, then you have two numbers that sum to 16. So the question becomes: What's the smallest n s.t. if you put n balls into 4 boxes you have to have some box with at least two balls. Using PGG this means the smallest n s.t. $\lceil n/4 \rceil \geq 2$, which is $n = 5$.

24.

We've seen in class, using PP, that there must be a triangle of mutual friends or mutual enemies in any group of 6 people. The proof does not go through if $n = 5$ so let's reconstruct a graph of n nodes with no all-friends or all-enemies triangle by looking at the case where the proof breaks down. It's easy to draw such a graph. Here's one assignment. The "friends" edges are (A,B), (A,C), (B,D), (C,E), (D,E), and the "enemy" edges are the five remaining edges: (A,D), (A,E), (B,C), (B,E), (C,D). You can check that there's no all-friends or all-enemies triangle.

32.

The boxes are the possible numbers of neighbors for any computer. (A *neighbor* of computer C is a computer that C is connected to.) Therefore there are only $k = 5$ boxes, because for each computer C the number of computers that C is connected to is between 1 and 5. Since there are $n = 6$ computers, by PP you must have a box with two balls in it, i.e. there must be two computers with the same number of neighbors.

34.

This is similar to the exercise we did in the lecture. The boxes are printers, so $k = 4$. Using the same construction as we used in the lecture, you can satisfy the constraint using $4 + 4 * 4 = 20$ cables: Divide the computers into two groups, group A and group B, with four computers each. Connect each node in A to a unique printer (need 4 cables for that) and then connect each node in B to all printers (need $4 * 4$ cables for that).[3] The reason this works is the same as in the lecture example: If any $t \leq 4$ of computers in A need to print, there's still $4 - t$ free printers and therefore any $4 - t$ set of computers in B can print too. Now, why is 19 not enough? Let the printers be boxes, i.e. $k = 4$ and cables be balls, i.e. $n = 19$. By the "GPP-2" principle we described in class, if there are n balls and k boxes then there must be a box that have *at most* $\lfloor n/k \rfloor$ balls. In our case there must be a printer P that has at most $\lfloor 19/4 \rfloor = 4$ connections. Let L be the lucky set of computers that are connected to printer P. Therefore there's only at most 4 computers in set L. Therefore any set of four computers chosen among the remaining 6 computers has no connection to P, and hence has only at most 3 servers to print from. This violates the requirement that any four computers can directly access four different printers. Therefore 19 cables is not enough.

---

[3] If someone gives another construction for how to satisfy this constraint with 20 cables, let me know and you'll get bonus points!