Chris Piech
CS 109

October 12, 2018

Problem Set #1 Solutions

1. a. 26!. Any arrangement of 26 letters is possible.

b. $2 \times 25!$. Treat the Q and U as a single object. Permute the now 25 remaining objects. As a second step in the experiment (via the product rule), choose the order of the Q and the U.

2. a. $\frac{208!}{(4!)^{52}}$. This is a permutation with repetitions question. There are 208 cards, but this consists of one group of 4 indistinguishable cards for each of the 52 distinguishable types of card. So we need to divide out the (4!) ways of ordering each indistinguishable group.

b. Sum of the integers from 1 to 52 $= \frac{(52)(53)}{2} = 1378$. Let us consider each card having a value 1 through 52. Card 1 can be matched with any of the other 52 values; card 2 can be matched with 51 other values, since we have already counted the match of card 1 and card 2; and so on. This forms the upper triangle of a grid which shows all 52^2 pairs of card types (the lower triangle has pairs that already exist in the opposite order in the upper triangle).

Note that the answer is not $52^2 = 2704$, since the order of the cards in your hand does not matter. Nor is it $\binom{52}{2} = 1326$, which is the number of ways of choosing two cards without replacement. This is because we have multiple decks, and you could get a hand which has two Aces of Spades (for example). You can account for this by adding 52, however: $\binom{52}{2} + 52 = 1378$.

c. Sum of the integers from 1 to 20 $= \frac{(20)(21)}{2} = 210$. This is the same as part (b), but we only have 20 distinct good cards: 10, J, Q, K, and A of each of the four suits. We can also express this as $\binom{20}{2} + 20 = 210$.

3. $\binom{n-k+r-1}{r-1}$, where $k = \sum_{i=1}^{r} m_i$. If you consider that k requests must be allocated according to the constraints that the i-th server receives at least m_i requests, it leaves $n - k$ requests to distribute in the r servers. The number of ways to do this is the same as finding the number of solutions to $x_1 + x_2 + ... + x_r = nk$, where each of the $x_i \geq 0$.

4. $\sum_{j=0}^{k} \binom{j+n-1}{n-1}$. Since $\sum_{i=1}^{n} x_i \leq k$, we consider all possibilities $\sum_{i=1}^{n} x_i = j$, where $0 \leq j \leq k$. For any particular value j, the number of non-negative integer solutions to $x_1+x_2+\cdots+x_n = j$ is given by $\binom{j+n-1}{n-1}$. To count all possible vectors, we sum this quantity over all values of j from 0 to k.

Alternatively, an equivalent answer is: $\binom{k+n}{n}$. This answer comes from conceptually adding an extra element x_{n+1} to the vector to represent the unallocated values, and then counting the number of non-negative integer solutions to $x_1 + x_2 + \cdots + x_n + x_{n+1} = k$. Note that if we were to subtract x_{n+1} from both sides of that equation, we get $x_1 + x_2 + \cdots + x_n = k - x_{n+1}$, which is equivalent to $x_1 + x_2 + \cdots + x_n \leq k$, since x_{n+1} is a non-negative integer $\leq k$.

5. a. $\binom{n+m-2}{n-1} = \frac{(n+m-2)!}{(n-1)!(m-1)!}$. Regardless of the path, the robot must make $(m-1)$ right moves and $(n-1)$ up moves. We can compute the number of paths as the number of ways to choose the up moves out of the $(m-1)+(n-1)$ total moves. Or we can reason about distinguishable permutations: the moves can be made in any order, but right moves are indistinguishable from other right moves, and up moves are indistinguishable from other up moves.

b. $\binom{n+m-3}{n-1} = \frac{(n+m-3)!}{(n-1)!(m-2)!}$. This is the same as part (a), but with one less right move available to order.

c. $2(n-2)(m-2)$. There are two possible cases: [ups, rights, ups, rights] or [rights, ups, rights, ups].

- Case 1: [ups, rights, ups, rights]. Valid paths for this sequence will look like:

  D ⟨more ups⟩ R ⟨more rights⟩ D ⟨more ups⟩ R ⟨more rights⟩

  Of the $(n-1)$ up moves, two are fixed. We can place any number of the $(n-3)$ remaining up moves into the first slot for more up moves ($n-2$ options), and the rest will go in the third slot. Similarly, two of the $(m-1)$ right moves are fixed, so we can place any number of the remaining $(m-3)$ in the second slot ($m-2$ options). The total number of paths for this case is $(n-2)(m-2)$.
- Case 2: [rights, ups, rights, ups]. This case is exactly symmetric to part (a), but with the $(m-3)$ remaining rights going in the first and third slots, and the $(n-3)$ remaining ups going in the second and fourth slots. The number of paths for this case is also $(n-2)(m-2)$.

These are mutually exclusive, so we can sum up the number of paths from the two cases to get $2 \cdot (n-2)(m-2)$.

6. a. $\binom{13}{3} = 286$. Since you must invest the minimum in all the opportunities, you must invest $1+2+3+4 = \$10$ million. Then you have \$10 million left to invest in the 4 opportunities, which has the same number of possibilities as solutions to $x_1 + x_2 + \cdots + x_4 = 10$.

b. $\binom{13}{3} + \binom{13}{2} + \binom{14}{2} + \binom{15}{2} + \binom{16}{2} = 680$. First, you still need to consider all the cases where you invest in all 4 opportunities (same as in part (a)), then you can use a similar analysis from part (a) to consider the number of possibilities if you (i) didn't invest in company 1, (ii) didn't invest in company 2, (iii) didn't invest in company 3, and (iv) didn't invest in company 4, respectively. Summing all these possibilities gives the complete answer.

7. a. $\binom{4}{1}\binom{13}{5} \Big/ \binom{52}{5}$. Choose 1 of the 4 suits for the flush, then choose 5 of the 13 cards in that suit.

b. $\binom{13}{1}\binom{4}{2}\binom{12}{3}\binom{4}{1}^3 \Big/ \binom{52}{5}$. Choose 1 of the 13 card ranks for the rank of the pair. Then choose 2 of the 4 cards of that rank to form the pair. Then choose 3 of the remaining 12 ranks, and choose 1 of the 4 cards at each of those ranks to form the rest of the hand.

c. $\binom{13}{2}\binom{4}{2}^2\binom{44}{1} \Big/ \binom{52}{5}$. Choose 2 of the 13 card ranks for the ranks for the two pairs, and choose 2 of the 4 cards of each rank to form each pair. Then choose 1 of the 44 (=52-8) cards that does not share the same rank as one of the 2 chosen pairs to complete the hand.

d. $\binom{13}{1}\binom{4}{3}\binom{12}{2}\binom{4}{1}^2 \Big/ \binom{52}{5}$. Choose 1 of the 13 card ranks for the rank of the three of a kind, and choose 3 of the 4 cards of that rank to form the three of a kind. Then choose 2 of the remaining 12 ranks, and choose 1 of the 4 cards at each of those ranks to form the rest of the hand.

e. $\binom{13}{1}\binom{4}{4}\binom{48}{1} \Big/ \binom{52}{5}$. Choose 1 of the 13 card ranks for the rank of the four of a kind, and choose 4 of the 4 cards of that rank to form the four of a kind. Then choose 1 of the 48 (=52-4) cards that does not share the same rank as the four of a kind to complete the hand.

8. a. $\binom{6}{3}\binom{6}{2}\binom{4}{2}\binom{2}{2} \Big/ 6^6$. First, select the 3 different numbers (out of 6) that are each rolled twice. Then, in the 6 rolls there are 3 sets of 2 indistinguishable rolls (i.e., each number is rolled twice). Since the die rolls are distinct, there are 6^6 total outcomes for rolling the die 6 times.

b. $\binom{6}{1}\binom{6}{4}5^2 \Big/ 6^6$. First, we select 1 of the 6 numbers that will appear exactly 4 times (call this number x). Then, we select the 4 rolls (out of 6) where x is rolled. The other 2 rolls can be any one of the 5 other numbers on the die that are not x. Similar to part (a), there are 6^6 total outcomes for rolling the die 6 times.

9. a. $\frac{2^{n-1}}{n!}$. First, note that there are $n!$ orderings of the values 1 through n we insert into the BST. Now, the only way to produce a completely degenerate BST is to have every successive insertion be either the minimal or maximal value of the values remaining to be inserted. This means we have one of 2 choices for every successive element we insert into the BST (minimal or maximal), except for the last element inserted (since it is both the minimal and maximal remaining element). Since we insert n elements total, we have 2^{n-1} insertion orderings that can produce a completely degenerate BST.

b. $n = 9$ is the smallest value such that $\frac{2^{n-1}}{n!} < 0.001$. Specifically, $\frac{2^8}{9!} \approx 0.000705$. Note how rapidly the probability of producing a degenerate BST decreases with the number of elements inserted.

10. a. $1/n$. You can think of this as analogous to arranging the passwords linearly, with a numbering from 1 to n on the passwords. We want to determine the probability that the correct password is given the number k. This is $1/n$, since the number k is equally likely to be assigned to any of the n passwords.

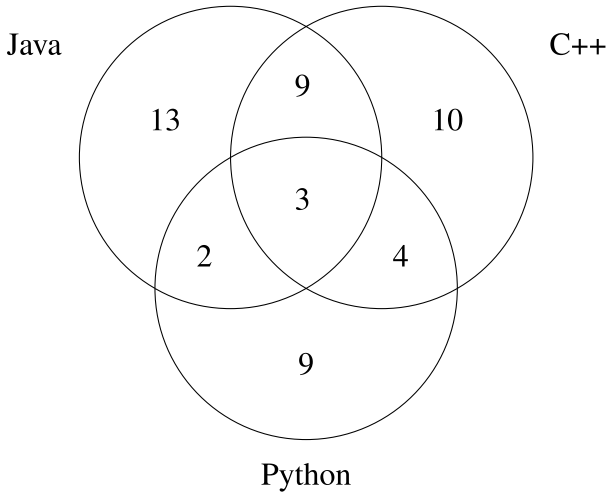
b. $\frac{(n-1)^{k-1}}{n^k}$. There are n^k ways of choosing any password for the first k attempts. Out of those, the ways of getting the right password for the first time on the k-th attempt are the number of ways of choosing any wrong password for each of the first $k-1$, which is $(n-1)^{k-1}$, times the number of ways of getting the right password on the k-th try, which is 1.

This can also be solved using independence. To successfully log in on exactly the k-th try, the hacker would have had to pick one of the wrong passwords (where the probability of picking one of the wrong passwords on a particular attempt is $\frac{n-1}{n}$) on each of the $k-1$ previous tries, yielding $\left(\frac{n-1}{n}\right)^{k-1}$. Then, on the k-th try, the hacker would have a $\frac{1}{n}$ probability of selecting the right password, giving the final answer.

11. a. $\frac{1}{2}$. We compute this by noting the probability that a student is in at least 1 programming class is $P(\text{Java} \cup \text{C++} \cup \text{Python}) = (27 + 26 + 18 - 12 - 5 - 7 + 3)/100 = \frac{1}{2}$. The probability that a student is not in any programming class is:

$$1 - P(\text{Java} \cup \text{C++} \cup \text{Python}) = \frac{1}{2}.$$

b. 32/100. We compute this by explicitly forming the Venn diagram of class enrollments (shown below), and then summing cell counts (13, 10, and 9) that are only a member of one set.



c. $149/198 \approx 0.75$. As shown in part (a), the probability that a student is not taking any programming class is $\frac{1}{2}$, which means there are 50 out of 100 students who are not taking any programming class. Now we randomly draw 2 students and determine the probability that *neither* one is taking a programming class as: $(50/100) \cdot (49/99)$. So, the probability that at least one of the drawn students is taking a programming class is:

$$1 - (50/100) \cdot (49/99) = 1 - 2450/9900 = 7450/9900 = 149/198.$$

12. $\binom{N}{k}\binom{M}{r-k}\Big/\binom{M+N}{r}$ or equivalently $\binom{r}{k}\binom{M+N-r}{N-k}\Big/\binom{M+N}{N}$. The first expression gives the answer by focusing on forming the first r bits of the received message. If we were to order all the N 1's with integers from 1 to N, we could select k of them to be in the first r bits of the message, yielding $\binom{N}{k}$ possibilities. Similarly, if we were to order all the M 0's with integers from 1 to M, we could select $(r - k)$ of them to be in the first r bits of the message, yielding $\binom{M}{r-k}$ possibilities. Multiplying, we get $\binom{N}{k}\binom{M}{r-k}$ total ways of arranging the first r bits, having exactly k 1's. The number of ways to form the first r bits of the message (with no constraints), is to think of numbering the $M + N$ bits with integers from 1 to $(M + N)$, and then choosing r of these integers to form the first r bits, yielding $\binom{M+N}{r}$ for the denominator.

Alternatively, we can get a mathematically equivalent answer to the problem (the second solution given above) by determining how to construct the whole binary string, so as to have k 1's in the first r bits. Consider the message being sent as have $M + N$ 'slots' to fill with 0's and 1's. The denominator is determined by choosing N of the $(M + N)$ slots to put 1's into.

The numerator is determined by choosing k of the first r slots to put 1's in, then choosing $(N - k)$ of the remaining $(M + N - r)$ slots to put 1's in. Once all slots for 1's are determined, the 0's uniquely fill the remaining unfilled slots.

13. $\binom{m}{k}(N-1)^{m-k}/N^m$. Choose k of the m strings to get hashed into the first bucket. Then each of the remaining $(m - k)$ strings can get hashed into any of the remaining $(N - 1)$ buckets. To get the denominator, recall that that all N^m arrangements are equally likely.

14. a. 66/144 = 11/24. One way we can solve this problem is to enumerate all the possibilities. Namely, if the first number generated is a 1 (with a 1/12 probability), then there are 11 possibilities for the second number to be greater (i.e., 11/12 probability). Similarly, if the first number is a 2, then there are 10 possibilities for the second number to be greater, etc. Summing all these possibilities yields: $(1/12)[11/12 + 10/12 + \cdots + 1/12] = 66/144$.

Another way to solve the problem is to use symmetry. Call the first number A and the second B. We start with the equation:

$$1 = P(A > B) + P(B > A) + P(A = B)$$

By symmetry, $P(A > B) = P(B > A)$, so we can rewrite the equation above as:

$$1 = 2[P(B > A)] + P(A = B)$$

Noting that $P(A = B) = 1/12$, we have:

$$\begin{aligned} 1 &= 2[P(B > A)] + 1/12 \\ 2[P(B > A)] &= 11/12 \\ P(B > A) &= 11/24 = 66/144 \end{aligned}$$

This is clearly the same as above, but does not require us to sum a series. Good times.