CS19: Solutions to Homework 3

Prepared by David Blinn and Amit Chakrabarti

February 5, 2006

You must demonstrate how you arrived at your final answers — i.e., you must show your steps — unless the problem statement makes an exception. You must also justify any steps that are not trivial. Simply writing down a final answer *will not earn any credit.* Please think carefully about how you are going to organise your answers *before* you begin writing.

The notation P$i.j$-k refers to Problem k from the list of problems after Section $i.j$ in your textbook. Thus, P1.2-4 refers to Problem 4 on page 17.

1. Solve P1.3-10.

   Solution: The difference between the two scenarios is that in the labeling situation, once we have chosen four disjoint three-element subsets, we may label these subsets in 4! ways. That is, choosing the sets does not tell us which set to label with which label. In the scenario in which we label, from the lesson on labeling and trinomial coefficients, we have $\binom{12}{3}\binom{9}{3}\binom{6}{3}\binom{3}{3} = \frac{12!}{3!3!3!3!}$ ways of assigning the labels. In the case in which we only choose subsets, the number of choices of the four sets is the number of labelings divided by 4!, namely $\frac{12!}{(3!)^4 4!}$. We can figure out the number of ways to choose three disjoint 4-element subsets from a 12-element set: $\frac{12!}{(4!)^3 3!}$.

2. By rearranging the letters in the word BULB it is possible to form 12 letter strings, as follows:

BULB	BLUB	BUBL	BLBU	BBLU	BBUL
LUBB	LBBU	LBUB	ULBB	UBBL	UBLB

   For each of the following words, how many letter strings can you form by rearranging the letters in the word?

   For each of the following 3 problems, we can apply the following logic: If there are no letter duplicates in a word, the number of letter strings we can form by rearranging the letters in the word is the number of ways we can permute the letters. So, for an n letter word with no duplicate letters, we could form $n!$ letter strings. In the case in which a word contains duplicate letters, taking all of the permutations of the letters will create some duplicate arrangements (strings that belong in the same equivalence class). If a letter appears m times in the word, then each arrangement has been listed $m!$ times because we have listed all permutations and there are $m!$ ways to permute the duplicates. Thus, for each letter that appears more than once in the word, if we divide by the number of ways of permuting the duplicate letters, we arrive at the proper number.

   (a) HATCH
   Solution: 5 total letters, 'H' appears 2 times. $\frac{5!}{2!} = 60$ We can think of this problem in terms of the refrigerator magnet letters problems presented in class in which we have two letter H's, one red and one blue. The order of the red and blue H's don't matter, and as a result we end up with equivalence classes of size 2!, which we divide by to get the number of equivalence classes.

(b) UNCOPYRIGHTABLE
Solution: 15 total letters, no duplicates. $15! = 1,307,674,368,000$

(c) SLEEPLESSNESSES
Solution: 15 total letters. 'L' apperas 2 times. 'S' apperas 6 times. 'E' appears 5 times. $\frac{15!}{2!6!5!} = 7,567,560$

3. Solve P1.3-19.

   Solution: False. $\binom{4}{2}$ is 6, but $\binom{2}{0} + \binom{2}{1} + \binom{2}{2}$ is 4. The correct statement is

$$\binom{n}{k} = \binom{n-2}{k-2} + 2\binom{n-2}{k-1} + \binom{n-2}{k}.$$

   The proof consists of applying the Pascal relation to both $\binom{n-1}{k-1}$ and $\binom{n-1}{k}$ and adding the results.

$$\begin{aligned}\binom{n}{k} &= \binom{n-1}{k-1} + \binom{n-1}{k} \\ &= \left(\binom{n-2}{k-2} + \binom{n-2}{k-1}\right) + \left(\binom{n-2}{k-1} + \binom{n-2}{k}\right) \\ &= \binom{n-2}{k-2} + 2\binom{n-2}{k-1} + \binom{n-2}{k}.\end{aligned}$$

4. Solve P1.4-5.

   Solution: Number the places around the table from 1 to $2n$. Then a seating gives a list of the $2n$ people. Once we decide the gender of the person in seat 1, we have $n!$ ways to seat that gender and $n!$ was to seat the other gender. So, we have $2 \cdot n! \cdot n!$ lists of people corresponding to seating arrangements. But then, there are $2n$ lists that correspond to the same circular arrangement. So, the number of ways of seating is $\frac{2 \cdot n! \cdot n!}{2n} = n! \cdot (n-1)!$

5. Figure out the sum of the *odd* binomial coefficients. To be precise, work out the sum

$$\sum_{\substack{i=1 \\ i \text{ odd}}}^{n} \binom{n}{i} = \binom{n}{1} + \binom{n}{3} + \binom{n}{5} + \cdots.$$

   If you are stuck for ideas, first try computing the sum for a few small values of n and see if that gives you a hint. If you are still stuck, look at P1.3-17.

   Solution: Consider the sum $\sum_{k=0}^{n} \left(\binom{n}{k}(-1)^k\right)$. Notice if we expand this out we get:

$$\sum_{k=0}^{n} \left(\binom{n}{k}(-1)^k\right) = \binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \binom{n}{3} \cdots.$$

   We see that the even terms are all positive and the odd terms are all negative. From the binomial theorem, we have that $\sum_{i=0}^{n} \binom{n}{i} x^{n-i} y^i = (x+y)^n$. If we apply this to the sum we are considering:

$$\sum_{k=0}^{n} \left(\binom{n}{k}(-1)^k\right) = \sum_{k=0}^{n} \left(\binom{n}{k} 1^{n-k}(-1)^k\right) = (1-1)^n = 0.$$

   Thus, in a sum of binomials, the sum of the even terms must be equal to the sum of the odd terms. From problem 6, we know that $\sum_{k=0}^{n} \binom{n}{k} = 2^n$. Thus, the sum of the odd binomails must be half, this, or $\frac{2^n}{2} = 2^{n-1}$.

6. Many equations involving binomial coeffients can be proven true in two ways. An *algebraic proof* simply uses the formula for $\binom{n}{k}$ on both sides of the equation and simplifies, or perhaps it uses other algebraic facts, such as the binomial theorem. A *combinatorial proof* interprets each side as counting a certain set of objects in two different ways. Here is an example.

   Theorem: $\binom{n}{k}\binom{n-k}{j} = \binom{n}{j}\binom{n-j}{k}$.

   Algebraic Proof: Using the formula for binomial coefficients on the left hand side, we have

$$\binom{n}{k}\binom{n-k}{j} \;=\; \frac{n!}{k!\cancel{(n-k)!}} \times \frac{\cancel{(n-k)!}}{j!(n-k-j)!} \;=\; \frac{n!}{k!j!(n-k-j)!}\,,$$

   and using it on the right hand side, we have

$$\binom{n}{j}\binom{n-j}{k} \;=\; \frac{n!}{j!\cancel{(n-j)!}} \times \frac{\cancel{(n-j)!}}{k!(n-j-k)!} \;=\; \frac{n!}{j!k!(n-j-k)!} \;=\; \frac{n!}{k!j!(n-k-j)!}\,.$$

   Thus, we see that the two sides are equal. QED.

   Combinatorial Proof: Let $S = \{1, 2, 3, \ldots, n\}$. The left hand side counts the number of ways of choosing a k-element subset of S followed by a j-element subset of the $n-k$ elements not previously chosen. In other words, it counts the number of pairs (A, B) where $A, B \subseteq S$, $|A| = k$, $|B| = j$ and $A \cap B = \emptyset$. Such a pair (A, B) can also be formed by choosing B first and then choosing A disjoint from B. Counting the pairs this way gives us the right hand side.

   Thus, we see that the two sides are equal. QED.

   For each of the following equations, give an algebraic *as well as* a combinatorial proof.

   (a) $\displaystyle\sum_{k=0}^{n}\binom{n}{k} \;=\; 2^n.$

   Hint for the algebraic proof: don't expand using factorials; use the binomial theorem.

   Solution:

   Algebraic Proof: First of all, note that $\sum_{k=0}^{n}\binom{n}{k} = \sum_{k=0}^{n}\left(\binom{n}{k}1^{n-k}1^k\right)$. Now, by the binomial theorem, we have $\sum_{k=0}^{n}\left(\binom{n}{k}1^{n-k}1^k\right) = (1+1)^n = 2^n$. QED.

   Combinatorial Proof: On the left side of the equation, we are summing the number of subsets of size $1, 2, \ldots, n$ that can be constructed from an n element set. This is the same as calculating the size of the power set of an n element set which we from class we know to be 2^n. QED.

   (b) $\binom{n}{k}\binom{k}{j} \;=\; \binom{n}{j}\binom{n-j}{k-j}.$

   Hint for the combinatorial proof: consider a subset of a subset.

   Solution:

   Algebraic Proof: Using the formula for binomial coefficients on the left hand side, we have

$$\binom{n}{k}\binom{k}{j} \;=\; \frac{n!}{\cancel{k!}(n-k)!} \times \frac{\cancel{k!}}{j!(k-j)!} \;=\; \frac{n!}{(n-k)!j!(k-j)!}\,,$$

   and using it on the right hand side, we have

$$\binom{n}{j}\binom{n-j}{k-j} \;=\; = \frac{n!}{j!\cancel{(n-j)!}} \times \frac{\cancel{(n-j)!}}{(k-j)!(n\cancel{-j}-(k\cancel{-j}))!} \;=\; = \frac{n!}{(n-k)!j!(k-j)!}\,.$$

Thus, we see that the two sides are equal. QED.

Combinatorial Proof: One manner of phrasing the left side is that it describes the number of ways of choosing a k element subset (call it K) from an n-element set and then choosing a j element sub-subset (call it J) such that $J \subseteq K$.

Notice that in doing the above we may have left some elements in K out of J. The number of these "left out" elements is $k-j$. Thus, an alternate way of doing the same thing is to first choose the j elements of J from the n element set, and then choosing the remaining $k-j$ "left out" elements to add to J, forming K. This is precisely the situation the right side of the equation describes, and thus the two sides are equal. QED.

(c) $\displaystyle\sum_{k=0}^{n} \binom{n}{k}\binom{n}{n-k} \;=\; \binom{2n}{n}$.

Hint for the algebraic proof: use the binomial theorem to analyze the expression $(1+x)^n(1+x)^n$; what is the coefficient of x^n after you multiply out and collect like terms?

Hint for the combinatorial proof: consider choosing n persons out a group of n men and n women.

Solution:

Algebraic Proof: Consider the expression $(1+x)^n(1+x)^n$. If we combine the two terms by adding the exponents, we have the expression $(1+x)^{2n}$. If we apply the binomial theorem, we see that the coefficient for the x^n term should be $\binom{2n}{n}$.

Now suppose we apply the binomial theorem to the expression $(1+x)^n(1+x)^n$ by considering the terms individually. We see that we get the expression

$$\left[\binom{n}{0}x^0 + \binom{n}{1}x^1 + \ldots + \binom{n}{n}x^n\right] \times \left[\binom{n}{0}x^0 + \binom{n}{1}x^1 + \ldots + \binom{n}{n}x^n\right].$$

If we multiply out, collect like terms and examine those that combine to give the x^n term, we see that we get $\binom{n}{0}\binom{n}{n}x^0x^n + \binom{n}{1}\binom{n}{n-1}x^1x^{n-1} + \binom{n}{2}\binom{n}{n-2}x^2x^{n-2} + \cdots + \binom{n}{n}\binom{n}{0}x^nx^0$. If we write the coefficient as a summation, we see that the coefficient of x^n is $\sum_{k=0}^{n}\binom{n}{k}\binom{n}{n-k}$.

We have worked out the coefficient of x^n in the polynomial $(1+x)^n(1+x)^n$ in two different ways. The two answers must be equal. Therefore $\sum_{k=0}^{n}\binom{n}{k}\binom{n}{n-k} \;=\; \binom{2n}{n}$.

Combinatorial Proof: Suppose we have to select an n element subset from a group of $2n$ elements. The number of ways in which we can do this is $\binom{2n}{n}$. Another way to perform the same task is to split our set of size $2n$ into two separate sets of size n. We can then choose k elements from the first set and $n-k$ elements from the second set. This corresponds to the process of choosing as described on the left side of the above equation, and thus the two sides are equal.

7. Solve P1.4-15. If you don't know `C++` syntax, feel free to write your improved code in the language of your choice. If you know what "pseudocode" means, you may use that too.

   Solution: Note that there are many possible improvements that could be made to speed up the code. Knowing that $\binom{n}{k} = \binom{n}{n-k}$ you might convert the one to the other if it is to your advantage, although this will only speed up the runtime in certain cases. Alternatively, you might use a technique called *memoization* to prevent the recomputation of certain terms of Pascal's triangle that takes place in the original code.

   The solution below simply takes advantage of the fact that

$$\binom{n}{k} \;=\; \frac{n^{\underline{k}}}{k!} \;=\; \frac{n}{1} \times \frac{n-1}{2} \times \cdots \times \frac{n-k+1}{k}.$$

```
int pascal(int n, int k)
{
  if (n < k)
    {
      cout << "error: n<k" << endl;
      exit(1);
    }

  int n_falling_k = 1, k_fact = 1;
  if (k > n/2)
      k = n - k;
  for (int i = n; i > (n - k); i--)
      n_falling_k *= i;
  for (int i = 1; i <= k; i++)
      k_fact *= i;

  return (n_falling_k / k_fact);
}
```

We will learn a better method for analyzing algorithms such as these in the future, but for the moment, note that the original slow code makes many recursive calls until it reaches a base case when either k is 1 or n and k are equal. Thus, it must make at least $\binom{n}{k}$ operations to perform the calculation (and in most caess many more). The new code, however, just calculates one falling factorial and one factorial in linear time, and uses the results to compute the answer. Just note that n and k are generally much smaller than $\binom{n}{k}$.