1. A hat check person discovers that n people's hats have been mixed up and returns these hats to the owners at random.

1. (a) In how many ways can the hats be returned so that the owners get someone else's hat?
2. (b) What proportion of the total number of distributions do these represent? Justify your answer.

2. Using P for pennies, N for nickels, D for dimes, and Q for quarters, write the symbolic series for making change using from zero to five of each kind of coin. What should you substitute for each letter so that the coefficient of x^n in the result is the number of ways to make n cents using from zero to five of each coint? What is the polynomial that results?

3. Write down the symbolic series and then the corresponding generating function for the number of ways to choose an odd number of apples and a multiple of 3 of tangerines from unlimited supplies.

4. In class, we considered the number of ways to make change for a dollar using nickels, dimes, and quarters. Extend the method to allow for pennies as well. (Note that we may assume that the number of pennies is always a multiple of 5). Also, look at the discussion in Section 3.3 of Bogart. In this case, how many ways can we make change for a dollar?

5. What is the generating function for the number of partitions of an integer into parts all of which are even numbers?

6. Use generating functions to solve the recurrence relation $a_{n+2} = 4a_{n+1} - 4a_n$, assuming $a_0 = a_1 = 1$.

7. A merge sort of a list of numbers can be described as follows. If the list has only one element, do nothing. Otherwise, split the list in half, apply merge sort to each half, then merge the two sorted lists in increasing order. Let a_n be the number of comparisons made by a merge sort on an n-element list. For $n = 1, 2, 4$, figure out by experiment how many comparisons you use. Assuming n is a power of 2, weite a recurrence relation for the numbers a_n. Since this recurrence involves $a_{n/2}$, it is not linear, and the merging keeps it from being homogenous. There is a solution to this recurrence involving $n \log_2 n$. One way to find it is to make the substitution $n = 2^k$. Make this substitution, solve the resulting recurrence, and convert back from k to n to get a formula for a_n. This kind of recurrence frequently arises in analyzing many of the "divide and conquer" algorithms in computing. (Quick sort is another good example of this).