This lecture explains the homework assignments 11 and 12.

The two assignments ask you to write programs that can shuffle cards.

The purpose of shuffling is to mix cards so that it is difficult to predict which card may appear next. The two assignments consider a particular method of shuffling called Riffle shuffling. This method has three steps. The first step divides a deck of cards into two parts. The two parts are held by left hand and right hand. The third step releases the cards so that they interleave. Usually the cards are released by the thumbs.

These two assignments are related. In homework 11, your program needs to shuffle a deck of cards only once. In homework 12, your program needs to shuffle the cards again. The assignments consider all possible shuffling results under two restrictions. The first restriction is that each hand must have at least one card when the original deck of cards is divided into two parts. The second restriction is that after the cards are divided into two parts, the order within each part must be preserved.

In these two assignments, we will consider only the numbers. We will not distinguish suits: diamonds, clubs, hearts, or spades.

Let’s consider the first restriction. Consider the situation when we have 8 cards. We can divide the cards in different ways. We can start with only one card held by the left hand and the remaining 7 cards on the right hand. Another possible way to divide the cards is to have two cards on the left hand and the remaining 6 cards on the right hand. The number of cards on the left hand can increase until there are 7 cards on the left hand and only one card on the right hand.

In general, if there are N cards originally, there are N minus 1 ways to divide the cards. The left hand may have 1, or 2, or 3, or N minus 1 cards.

Next, let’s consider how to interleave the cards. We consider the simplest case. There is only one card on the left hand and the right hand has 7 cards.

The only card may be put on the top of the seven cards, inserted between two of the seven cards, or at the bottom of the seven cards. Please notice that the order of the seven cards will not change.

Next we consider a more complex case. The left hand has 3 cards and the right hand has 5 cards. The cards from the two hands will interleave but the order of the cards on the left hand will not change. The 3 cards on the left hand have ei above 2 and 2 above 3. After shuffling, ei must be above 2, and 2 must be above 3. It is possible to have cards from the right hand inserted between ei and 2, or between 2 and 3. However, ei must always be above 2, and 2 must always be above 3.

Similarly, the order of 4, 5 , 6, 7, and 8 must not change. 4 must always be above 5. 5 must always be above 6.

This is a property of Riffle shuffling.

After understanding this restriction, we can select the card at the bottom. The card at the bottom must be either 3 or 8 because 3 is the bottom card from the left and 8 is the bottom card on the right hand.

If you look carefully, this is an example of a branch. The program has two choose one of the two possible cases. As explained earlier, when a program has to consider branches, recursion may be a good method to solve the problem.

Let’s look into this problem even more deeply. After dividing the cards into the two parts, we want to interleave the two parts. The first part has three cards: ei, 2 , and 3. The second part has 5 cards: 4, 5, 6, 7, and 8.

The bottom card in the result can be either 3 or 8. If 3 is chosen, the problem becomes interleaving 2 cards with 5 cards. If 8 is chosen, the problem becomes interleaving 3 cards with 4 cards.

Do you notice the similarity of the new problems with the original problem?

This problem has the tree essential components of recursion.

What is the stop condition? When one hand has no card left, this is the stop condition.

What is the change? Each time, we take one card from either left hand or right hand. Thus, the number of cards decreases by one in one of the two hands.

The recurring pattern is that after taking one card, the new problem is similar to the original problem but the total number of cards decreases by one.

This method considers the bottom instead of the top because I assume the top card has index zero. When the index reaches zero, we know only one card is left. This makes the problem slightly easier than going from the top to the bottom.

The slide shows the concept how to interleave the cards. Please be aware that this is not necessarily working code. This slide explains the concept.

This function takes six arguments: first, the cards on the left hand; second, the cards on the right hand; third, the destination cards that may come from either left hand or right hand; the other three arguments are the indexes for the three decks of cards.

This function has two choices: For the first choice, it picks a card from the left side. When this occurs, the number of cards from the left side decreases by one. The index for the destination also decrease by one. Here, we use decreases instead of increase because we start from the bottom. Thus, the indexes will decrease toward zero.

If the function choose one card from the right hand, then the index for the left hand is unchanged. The index for the right hand decreases by one. The index for the destination also decreases by one.

What is the original values of the three indexes? For the left and the right hands, the starting indexes are determined by the number of cards assigned by the divide stage. For the destination, the index will start at the value of the total number of cards.

Riffle shuffle actually has serious problems. Some orders cannot be generated by shuffling only once. Consider this simple example of three cards: ei, 2, and 3.

We can divide the cards into ei in the left hand. The right hand has 2 and 3. ei may be above 2 and 3, between 2 and 3, or below 2 and 3.

Next we consider the other way to divide the cards. The left hand has ei and 2 in one group. The right hand has only card 3. Card three may be below ei and 2, between ei and 2, or above ei and 2.

If we examine the produced orders, we notice that one possible order is missing. Riffle shuffle cannot generate the order when 3 is above 2 and 2 is above ei. .

As the number of cards increases, the problem because more serious.

Let’s consider a general case when there are N cards originally. The cards are divided into k cards on the left hand and n minus k cards on the right hand.

The N cards can be arranged in different orders but the orders on the left hand and on the right hand must be preserved.

Thus, there number of possible order is this:

The numerator is N factorial. The denominator is the product of K factorial and n minus k factorial. .

The value of K can be 1, or 2, or 3, up to N minus 1. The number of output using raffle shuffling is the sum for K equal to 1, or 2, or 3, up to N minus 1.

What is the value of this sum?

X plus y to the power of N can be written as the sum of this term: the numerator is N factorial.

The denominator is the product of K factorial and n minus k factorial. The value of k is 0, 1, 2, up to N. .

If we set x and y to both 1, then the left size is 2 to the power of N. .

If we remove the first and the last terms, then the sum is 2 to the power of N minus 2.

What does this mean?

As the number of cards increases, Riffle shuffle will miss more and more orders of cards.

When there are N cards, there are n factorial possible orders.

Two to the power of N is much smaller than N factorial as N increases. When N is 4, N factorial is 24. Two to the power of 4 minus 2 is only 14. This 14 possible orders include some repeated orders. Thus, the actually possible order is less than 14. When N is 6, two to the power of 6 minus 2 is 62. The factorial of 6 is 720.

If you want to use Riffle shuffling, shuffling once is definitely insufficient. You must shuffle multiple times.

How many times do you need to shuffle? This is actually a complex mathematics problem and is beyond the scope of E C E. 264.

You can find some additional information about card shuffling in the handout in Brightspace.

These two assignments really need thinking and design before coding. When you do homework 11, you should be aware of homework 12’s requirements.

You absolutely must design before writing code. You must have a plan before you write code. If you start coding without a plan, you will not finish the assignments.

The sample solution for homework 12 has 124 lines, including blank lines and comments. If your solution has more than 500 lines, your solution is probably wrong.

It is extremely important that you think about your solution before writing code.