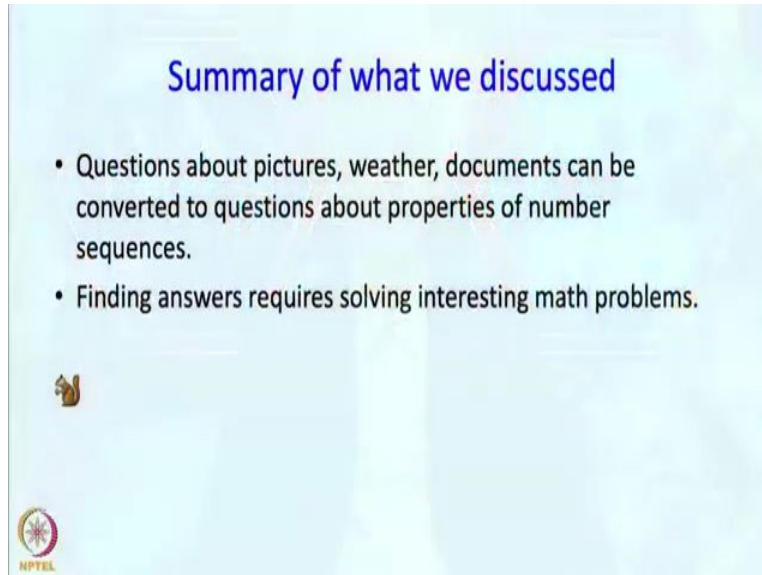


An Introduction to Programming through C++
Professor Abhiram G. Ranade
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Lecture No. 2 Part - 2
Problem Solving using Computer
Problem Solving and Algorithms without Computers

(Refer Slide Time: 0:23)




Welcome back! In the previous segment, we discussed about following - we said that questions about pictures, weather, documents can be converted to questions about properties of certain number sequences. And we also said that finding answers requires solving interesting math problems.

(Refer Slide Time: 0:45)

A historical remark

- Computers are used to solve problems, but problem solving is not new, humans have been solving numerical problems for millenia
 - Astronomical calculations for navigation
 - Land records/geometry
 - Poetry raised some interesting computational problems and some were solved very elegantly by ancient Indian mathematicians
 - Last 500 years: development of differential equations to analyze structures, weather, ...
- “Solving problems” = deciding what operations to perform to calculate the required answer
- Algorithms = A precise description of the operations needed to solve a problem.



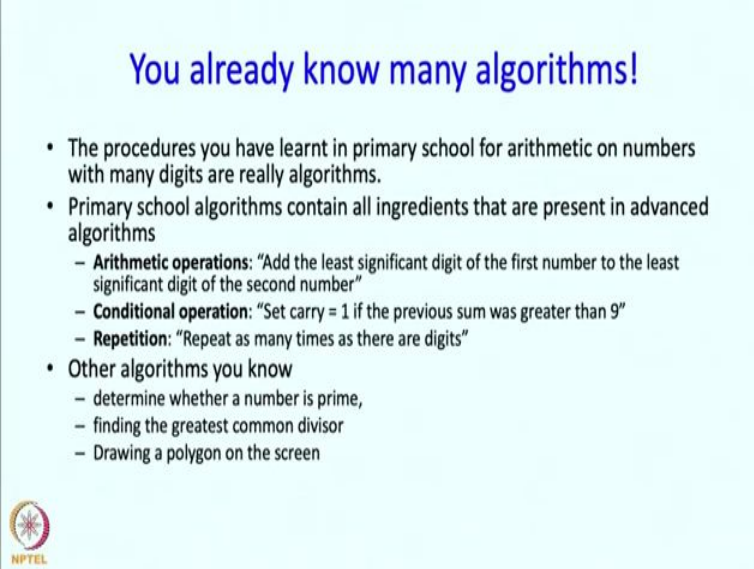
Now, I am going to begin with a historical remark in this segment. So, computers are used to solve problems. We know that, we see computer solving problems. But it is useful to remember that problem solving is really not new, human beings have been solving problems, numerical problems for thousands of years.

For example, astronomical calculations were done really thousands of years ago and astronomical calculations are quite sophisticated. They were done for navigation and of course, they were done for purposes of astrology as well. Land records were maintained and for that interesting geometric calculations had to be done. Turns out that in ancient India poetry raised some interesting computational problems and these were solved rather elegantly.

In the last 500 years, we have seen the development of differential equation which are being used to analyze structures, weather and lots of things. So these things really are problem solving and they deal with numerical problem solving and they are used on computers. But all of these has happened well before modern computers were anywhere in the picture. Okay, so what does the phrase “solving problems” mean? Well it means deciding what operations to perform to calculate the required answer and as I said, human beings have been solving problems or deciding what


operations to perform to calculate the required answer for quite some time. And an algorithm is simply a precise description of the operations needed.

(Refer Slide Time: 02:38)



You already know many algorithms!

- The procedures you have learnt in primary school for arithmetic on numbers with many digits are really algorithms.
- Primary school algorithms contain all ingredients that are present in advanced algorithms
 - **Arithmetic operations:** "Add the least significant digit of the first number to the least significant digit of the second number"
 - **Conditional operation:** "Set carry = 1 if the previous sum was greater than 9"
 - **Repetition:** "Repeat as many times as there are digits"
- Other algorithms you know
 - determine whether a number is prime,
 - finding the greatest common divisor
 - Drawing a polygon on the screen

 HPTTEL

Now, this might be a new term and you might think that it is a term associated with computers and you may think it as an alien term but actually it is not. I want to assure you that you already know many algorithms. So, you are already half-way there into learning the subject of computer programming. So the procedures you have learned in primary schools for arithmetical numbers, say, addition of 2 numbers with many digits are really algorithms.

Primary school algorithms, say on arithmetic, contain all the ingredients and that is really all that is important when you talk about solving these other more fancy problems. So, what does an algorithm contain? Well, it requires you to perform arithmetic operation. Your primary school algorithm for doing addition required you to do something like this - "Add the least significant digit of the first number to the least significant digit of the second number". This was what you needed to do in order to do addition.

There was also a conditional operation. So there was going to be a carry. So carry was 1 if the previous sum was greater than 9. So these conditional operations also appear in computer algorithms. And there is also repetition, so if you are adding one 10 digit number to another 10

digit number you will repeat this basic step 10 times and on a computer also, you need to repeat steps or you need to repeat a sequence of steps.

And really you do not need to do much more than this. You may do a lot of calculations, you may have to multiply numbers or do something, maybe something slightly different. But really these are the three kinds of steps that you are going to be doing when you execute a computer program. And algorithms is simply a description of such steps.

In addition to arithmetic you know actually many other rather sophisticated algorithms. So you know whether a number is prime. You know how to find the greatest common divisor of 2 numbers. This is a really pretty algorithm invented, say 2000 years ago and it is still really useful. In fact, we will write a program to implement this algorithm on a computer. You can draw a polygon and drawing a polygon is also an algorithm. It is a procedure. It is a very very clearly stated procedure.


So I should really point out that you do not need to worry only about numbers when we talk about algorithms, okay. Drawing is also a systematic procedure drawing for drawing is also an algorithm. Well you might say that look a systematic procedure for anything say making tea is also an algorithm.

But let us not worry about that. We are going to be concerned with numbers and it is vital that you understand what algorithms or numbers are, and my contention is that you already understand that. Something like what you did in order to add two multi-digit numbers is an algorithm. And an algorithm nothing really very much more. It really contains operations of that kind arithmetic operations, conditional operations, repetitions.

(Refer Slide Time: 6:33)

Programs

- Algorithms written in precise syntax/language.
- C++ is one such language.
- Other languages: Fortran, Basic, Lisp, ...
- All these languages can be used to specify arithmetic operations, conditional execution, and repetition.
 - You have already seen how to specify repetition – use the repeat statement!




So what is a program? A program is simply an algorithm written in a precise syntax or a precise language. C++ is only one such language. There are other such languages like Fortran, Basic, Lisp. All these languages can be used to specify arithmetic operation, conditional execution and repetition. You have already seen how to specify repetition, use the repeat statement. So this is actually a fairly important crucial statement.

(Refer Slide Time: 7:12)

Exercise

- Write down the algorithm for multiplying an n digit number by another n digit number that you learned in primary school.
 - Break up the description into parts, e.g. in the first part we do multiplications and in the second we do additions.
 - To describe what operations you perform, it might be useful to give names e.g. Let Q_i , R_i denote the i^{th} least significant digits of the multiplicand and the multiplier.
 - Are there phases in the algorithm in which you do similar operations? If yes can you say what happens in “ith phase”?




So some thought exercises for you. So try to write down the algorithm you learned whatever in standard 3 for multiplying an n-digit number by another n-digit number okay. Well the teacher probably did not tell you the algorithm for multiplying n-digit numbers, but just showed you how to do the multiplication. And by now you know how to do it for an arbitrary number or digits. How are you going to write that? Well, you should break up the description into parts. So maybe you will say that in the first part we are going to do some multiplications and in the second part we are going to do some additions. Then, in order to describe what operations to perform, you may want to give some names.


So you may say that let Q_i and R_i denote the i th least significant digits of the multiplicand and the multiplier and then you may say that look let us multiply Q_i by R_j or something like that. Then you may also want to ask are there phases in the algorithm in which you do similar operations? If so, can you say what may happen in the i th phase? So what might happen in the i th phase might depend on i , but, if you can find a pattern, then you have really captured the essence. Then you will be in a much better position to write it down nicely and indeed convert this into a program.

(Refer Slide Time: 8:52)

What we discuss



- Notion of problem solving, algorithms is old
- An algorithm specifies a sequence of operations
 - May contain arithmetic operations
 - May contain operations to be executed conditionally
 - May ask for sequence of operations to be repeated
- You already know many nice algorithms
 - Arithmetic on numbers, matrices, root finding...
 - Very useful for programming
- Programs are formal descriptions of algorithms, in specific languages.



Okay, so, what have we discussed in the segment? We have discussed that the notion of problem solving and algorithms is old, literally thousands of years old. And sophisticated algorithms were present thousands of years ago. Euclid's GCD algorithm is one example. One really fantastic example is the square root finding algorithm that ancient Babylonians knew.

What is an algorithm? It is a specification. A precise specification of a sequence of operations. It may contain arithmetic operations. It may contain operations to be executed conditionally. It may ask for a sequence of operations to be repeated. And you should remember, you should always remember in this course, that you really know algorithms. So you do not have to be worried. Of course, you do not have to be scared of computers. You do not have to be scared of algorithms.

The algorithms that you know are arithmetic on numbers, primary school, matrices this is what you learnt in standard 11th and 12th, root finding that is also something that you might have learned in high school or 11th and 12th. This knowledge that you have acquired in your schooling is going to be very useful for programming and as I said, you are already halfway there as far as being ready to program because you know algorithms already. Then we said that the programs are very formal, very precise descriptions of algorithms in specific languages having specific syntax.

So we stop this segment over here and continue later.