

Algorithmic Thinking

Department of Computer Science
City University of New York, Graduate Center

Lecture : Algorithmic Thinking

September 20 , 2010

Outline

- 1 Algorithmic Thinking
 - What is a program?
 - Possible Errors
 - Recipe Analogy

Outline

- 1 Algorithmic Thinking
 - What is a program?
 - Possible Errors
 - Recipe Analogy

What is a program

- Program - series of instructions that specify exactly what the computer is supposed to do.
- Computers are not people.
- We can tell a person to go and buy a loaf of bread.
- But if we were instructing a robot, we might have to be much more specific:

Steps:

- Specify how much bread costs
- Check that there is enough money
- If not get enough money to pay for the bread
- Give store address and exact directions how to get there
- Send to purchase the bread

What is a program

- Program - series of instructions that specify exactly what the computer is supposed to do.
- Computers are not people.
- We can tell a person to go and buy a loaf of bread.
- But if we were instructing a robot, we might have to be much more specific:

Steps:

- Specify how much bread costs
- Check that there is enough money
- If not get enough money to pay for the bread
- Give store address and exact directions how to get there
- Send to purchase the bread

What is a program

- Program - series of instructions that specify exactly what the computer is supposed to do.
- Computers are not people.
- We can tell a person to go and buy a loaf of bread.
- But if we were instructing a robot, we might have to be much more specific:

Steps:

- Specify how much bread costs
- Check that there is enough money
- If not get enough money to pay for the bread
- Give store address and exact directions how to get there
- Send to purchase the bread

What is a program

- Program - series of instructions that specify exactly what the computer is supposed to do.
- Computers are not people.
- We can tell a person to go and buy a loaf of bread.
- But if we were instructing a robot, we might have to be much more specific:

Steps:

- Specify how much bread costs
- Check that there is enough money
- If not get enough money to pay for the bread
- Give store address and exact directions how to get there
- Send to purchase the bread

What is a program

- Program - series of instructions that specify exactly what the computer is supposed to do.
- Computers are not people.
- We can tell a person to go and buy a loaf of bread.
- But if we were instructing a robot, we might have to be much more specific:

Steps:

- Specify how much bread costs
- Check that there is enough money
- If not get enough money to pay for the bread
- Give store address and exact directions how to get there
- Send to purchase the bread

What is a program

- Program - series of instructions that specify exactly what the computer is supposed to do.
- Computers are not people.
- We can tell a person to go and buy a loaf of bread.
- But if we were instructing a robot, we might have to be much more specific:

Steps:

- Specify how much bread costs
- Check that there is enough money
- If not get enough money to pay for the bread
- Give store address and exact directions how to get there
- Send to purchase the bread

Steps of a program

In order to use a program to solve a problem, we go through a series of steps, (software life-cycle).

Steps

- Determine how to solve the problem and come up with an algorithm.
- An algorithm is a series of steps that can be followed to solve the problem.
- Use a special program called a compiler to "translate" the program written in the high-level programming language to machine language.
- Run (or execute) the program. At this point, we may find other errors and we may have to correct the program again.
- Debug the program. Test it and see if there are any mistakes. Fix errors and test again.

Steps of a program

In order to use a program to solve a problem, we go through a series of steps, (software life-cycle).

Steps

- Determine how to solve the problem and come up with an algorithm.
- An algorithm is a series of steps that can be followed to solve the problem.
- Use a special program called a compiler to "translate" the program written in the high-level programming language to machine language.
- Run (or execute) the program. At this point, we may find other errors and we may have to correct the program again.
- Debug the program. Test it and see if there are any mistakes. Fix errors and test again.

Steps of a program

In order to use a program to solve a problem, we go through a series of steps, (software life-cycle).

Steps

- Determine how to solve the problem and come up with an algorithm.
- An algorithm is a series of steps that can be followed to solve the problem.
- Use a special program called a compiler to "translate" the program written in the high-level programming language to machine language.
- Run (or execute) the program. At this point, we may find other errors and we may have to correct the program again.
- Debug the program. Test it and see if there are any mistakes. Fix errors and test again.

Steps of a program

In order to use a program to solve a problem, we go through a series of steps, (software life-cycle).

Steps

- Determine how to solve the problem and come up with an algorithm.
- An algorithm is a series of steps that can be followed to solve the problem.
- Use a special program called a compiler to "translate" the program written in the high-level programming language to machine language.
- Run (or execute) the program. At this point, we may find other errors and we may have to correct the program again.
- Debug the program. Test it and see if there are any mistakes. Fix errors and test again.

Steps of a program

In order to use a program to solve a problem, we go through a series of steps, (software life-cycle).

Steps

- Determine how to solve the problem and come up with an algorithm.
- An algorithm is a series of steps that can be followed to solve the problem.
- Use a special program called a compiler to "translate" the program written in the high-level programming language to machine language.
- Run (or execute) the program. At this point, we may find other errors and we may have to correct the program again.
- Debug the program. Test it and see if there are any mistakes. Fix errors and test again.

Steps of a program

In order to use a program to solve a problem, we go through a series of steps, (software life-cycle).

Steps

- Determine how to solve the problem and come up with an algorithm.
- An algorithm is a series of steps that can be followed to solve the problem.
- Use a special program called a compiler to "translate" the program written in the high-level programming language to machine language.
- Run (or execute) the program. At this point, we may find other errors and we may have to correct the program again.
- Debug the program. Test it and see if there are any mistakes. Fix errors and test again.

Steps of a program

In order to use a program to solve a problem, we go through a series of steps, (software life-cycle).

Steps

- Determine how to solve the problem and come up with an algorithm.
- An algorithm is a series of steps that can be followed to solve the problem.
- Use a special program called a compiler to "translate" the program written in the high-level programming language to machine language.
- Run (or execute) the program. At this point, we may find other errors and we may have to correct the program again.
- Debug the program. Test it and see if there are any mistakes. Fix errors and test again.

Compiler vs Interpreter Analogy - Interpreter

In this analogy, English = the SOURCE CODE. French = the MACHINE CODE. You can think of the French speaker's brain as the computer's CPU

- The English speaker employs an interpreter who translates the English sentences into French as they are spoken.
- The English speaker says a sentence in English, the interpreter hears it, translates it and speaks the sentence in French.
- Progress is slow: there are pauses between sentences as the translation process takes place.

Compiler vs Interpreter Analogy - Compiler

The English speaker writes down (in English) what needs to be said. The whole document is then translated into French, producing another piece of paper with the French version written on it. There are two factors to consider in this scenario:

- There is a slight delay at the start as the English document needs to be translated in its entirety before it can be read.
- The translated document can be read at any time after that, and at the normal speed at which the French-speaker reads.

Outline

- 1 Algorithmic Thinking
 - What is a program?
 - Possible Errors
 - Recipe Analogy

Two examples of bugs

- The Mars Climate Orbiter was lost in space in 1999. In the calculation of its route some programmers used the metric system while others used English units.
- A patriot missile failing to intercept a scud fired at US troops in 1991. The following is from an article:
Specifically, the time in tenths of second as measured by the system's internal clock was multiplied by 1/10 to produce the time in seconds.

Two examples of bugs

- The Mars Climate Orbiter was lost in space in 1999. In the calculation of its route some programmers used the metric system while others used English units.
- A patriot missile failing to intercept a scud fired at US troops in 1991. The following is from an article:
Specifically, the time in tenths of second as measured by the system's internal clock was multiplied by 1/10 to produce the time in seconds.

Two examples of bugs

- The Mars Climate Orbiter was lost in space in 1999. In the calculation of its route some programmers used the metric system while others used English units.
- A patriot missile failing to intercept a scud fired at US troops in 1991. The following is from an article:
Specifically, the time in tenths of second as measured by the system's internal clock was multiplied by 1/10 to produce the time in seconds.

An algorithm example

Finding the oldest person in a room

- Go through the people sequentially, keeping track of the oldest one so far.

The number of steps is proportional to number of people: $O(n)$.

- Have the people pair up, eliminate the younger of each pair.
- Keep pairing up and eliminating the younger ones of the pair until there is only one left.

This can be done in $\log n$ time, but requires carrying out comparisons in parallel.

Searching for a value, e.g. phone number

o Sequential search : $O(n)$

o Binary search : $O(\log n)$

Outline

- 1 Algorithmic Thinking
 - What is a program?
 - Possible Errors
 - Recipe Analogy

Recipe Analogy

- There may be many algorithms to solve a given problem.
- A recipe is a good analogy for a program. A recipe has certain components:

A recipe

- ingredients, variables, constants
- data types: integer, real numbers, character
- detailed step-by-step instructions i/o
- repeated actions (e.g. separate 5 eggs) : loops
- pre-defined action, functions, built-in and user-defined; parameters.
- decisions - (e.g. beat until stiff; bake until top browns)

Recipe Analogy

- There may be many algorithms to solve a given problem.
- A recipe is a good analogy for a program. A recipe has certain components:

A recipe

- ingredients, variables, constants
- data types: integer, real numbers, character
- detailed step-by-step instructions i/o
- repeated actions (e.g. separate 5 eggs) : loops
- pre-defined action, functions, built-in and user-defined; parameters.
- decisions - (e.g. beat until stiff; bake until top browns)

Thank you for your attention