2024-05-24 / Reading Comprehension / English for Software Development

Chapter 11. Happy Ending Not Guaranteed

[Preface; first paragraph]

Loops and recursion inject power into algorithms. Loops enable algorithms to _____ inputs of arbitrary size and complexity. Without loops, algorithms could deal only with small and simple cases of inputs. Loops get algorithms off the ground. They are to algorithms what wings are to airplanes: without wings, airplanes still can move, but their transportation potential cannot be _____. Similarly, there are some computations that algorithms can describe without loops, but the full power of computation can be realized _____ loops. Such power also needs to be controlled, however. As the *Groundhog Day* story illustrates vividly, a control structure over which one has no control is not a blessing but a curse. Goethe's poem, "The Sorcerer's Apprentice" may also come to mind (which may be better known in the United State through the Mickey Mouse sequence in the film Fantasia). The key to controlling a loop is to understand the condition that decides when it ends. Phil Connors can _____ end the loop he is in, leading to a happy ending. But this happens more as a _____ than by following a plan.

[No End in Sight; first to third paragraphs]

Suppose you are presented with an algorithm and have to decide whether the algorithm is worth running, that is, whether it produces a result in a finite amount of time. How would you __ about it? Knowing that the only source of nonterminating behavior is loops, you would first _____ all the loops in the algorithm. ____, for each loop you would try to understand the relationship between the termination condition and the instructions of the loop body, because the termination of a loop depends on the termination condition and how the values on which it depends are _____ by the loop body. This analysis will ____ you to decide whether a particular loop terminates and whether the algorithm has a chance of terminating. To _____ the termination of the algorithm, you have to perform this analysis for each loop in the algorithm.

Since the termination is such an important property of algorithms—it _____ algorithms that actually solve a problem from those that don't—it would be nice to know the termination property for any algorithm that one may want to use. However, performing a termination analysis for an algorithm is not an easy task, and it might take a _____ of time to carry out. Thus, we are tempted to automate this task, that is, to create an algorithm, say, *Halts*, that does the termination analysis automatically. Since we have many other algorithms for analyzing algorithms (for example, for parsing, see Chapter 8), this doesn't seem like such an outlandish idea.

_____, it is impossible to construct the algorithm *Halts*. It is not that this is too difficult at the moment or that computer scientist have not thought about the problem long and hard enough. No, we know that it is impossible *in principle* to craft an algorithm *Halts*. It is impossible now, and it will never be possible. This fact is often _____ the *unsolvability of the halting problem*. The halting problem is a major problem in computer science. It was _____ by Alan Tuning in 1936 as an example of an undecidable problem.

Sentences To Memorize

1. Loop and recursion are to algorithms what wings are to airplanes.
2. Computer science is no more about computers than astronomy is about telescopes.
3. Knowledge about undecidable problems shows us the scope and limits of computation