2024-04-19 / Reading Comprehension / English for Software Development

Chapter 7. Mission Intractable

[Tripping the Scale; first two paragraphs]

At the beginning of *Raiders of the Lost Ark*, Indiana Jones [ ] a cave in search of a precious golden idol, the target of his expedition. The idol rests on a scale that is designed to [ ] a series of deadly traps in case the idol is removed. To circumvent the protective mechanism, Indiana Jones replaces the idol [ ] a bag of sand that he hopes approximates the idol's weight. Alas, the bag is too heavy and triggers the traps, which [ ] a spectacular escape from the deadly cave.

Had Indiana Jones known the exact weight of the idol, he [ ] the bag with the precise amount of sand, and his exit from the cave would have been much less dramatic. But since he was probably not carrying around a scale, he needed to [ ] the weight in some other way. Fortunately, it is not too difficult to build a balance scale. Basically all you need is a stick; attach the sand bag to one end and the exact weight at [ ], and then fill the sand bag with sand until the stick is balanced. Assuming that Indiana Jones doesn't have an object that has the exact same weight [ ] the idol, he has to approximate the weight with a collection of objects. This doesn't seem like such a difficult problem. If the idol has an estimated weight of, say, 42 ounces (which is about 2.6 pounds or 1.2 kg) and Indiana Jones has six objects that weight 5, 8, 9, 11, 13 and 15 ounces, respectively, then after trying several combinations he will find that 5, 9, 13 and 15 ounce objects add up exactly 42 ounces.

[Shared Destiny; first two paragraph]

The generate-and-test algorithm for solving the weighting problem works only for relatively small inputs (less than 30 or so), which is fine for the specific problem Indiana Jones [ ]. But because of the [ ] of its runtime, the algorithm will never be able to [ ] inputs of size 100 or larger. This particular algorithm's exponential runtime does not mean that there can't be other, more [ ] algorithms with nonexponential runtimes. Currently, however, no such algorithm is known.

A problem that can only be solved by algorithms with exponential (or worse) runtime is called *intractable*. Since we only know algorithms with exponential runtime for the weighting problem, it may seem that it is intractable, but we don't really know [ ]. Maybe there is a nonexponential algorithm that just hasn't been [ ] yet. If we could prove that no such nonexponential algorithm can exist for the problem, then we would know the problem is intractable. A lower bound could provide us with [ ].