Python - Algorithm Justifications

In order to make claims about an Algorithm being efficient we need some mathematical tools as proof. These tools help us on providing a mathematically satisfying explanation on the performance and accuracy of the algorithms. Below is a list of some of those mathematical tools which can be used for justifying one algorithm over another.

- **Direct Proof** − It is direct verification of the statement by using the direct calculations. For example sum of two even numbers is always an even number. In this case just add the two numbers you are investigating and verify the result as even.
- **Proof by induction** − Here we start with a specific instance of a truth and then generalize it to all possible values which are part of the truth. The approach is to take a case of verified truth, then prove it is also true for the next case for the same given condition. For example all positive numbers of the form 2n-1 are odd. We prove it for a certain value of n, then prove it for the next value of n. This establishes the statement as generally true by proof of induction.
- **Proof by contraposition** − This proof is based on the condition If Not A implies Not B then A implies B. A simple example is if square of n is even then n must be even. Because if square on n is not even then n is not even.
- **Proof by exhaustion** − This is similar to direct proof but it is established by visiting each case separately and proving each of them. An example of such proof is the four color theorem.

Print Page