

1 Algorithms

An algorithm is a finite sequence of precise instructions for performing a computation or for solving a problem.

There are many properties of algorithms to keep in mind:

- Input - input values from a specified set
- Output - from each set of input values an algorithm produces output values
- Definiteness - the steps of an algorithm must be defined precisely
- Correctness - an algorithm should produce correct output values for each set of input values
- Finiteness - an algorithm should produce the desired output after a finite number of steps for any input
- Effectiveness - it must be possible to perform each step of an algorithm exactly and in a finite amount of time
- Generality - the procedure should be applicable for all problems of the desired form

The first algorithm to present is the linear search, or sequential search. This search begins by comparing a x and a_1 and continues with each a_n until a match is found.

The binary search works when the list is sorted. We first split the list into two smaller sublists of the same size, and keep splitting it up based on the comparison to the term to be found the middle term.

The bubble sort is one of the simplest sorting algorithms. It puts a list into increasing order by successively comparing adjacent elements.

The insertion sort begins with the second element and compares it with the the first element. Then the third element is compared with the first and then the second if it is larger than the first.

Many algorithms are designed to solve optimization problems. This means they want to maximize or minimize some parameter. Algorithms that make what seems to be the "best" choice at each step are called greedy algorithms.

An example would be the cashier's algorithm which makes changes using the fewest coins possible when change is made from quarters, dimes, nickels, and pennies.

The halting problem is a interesting problem. It asks whether there is a procedure that can input a computer program and determine whether the program will eventually stop.

The reason there isn't one is because you do not know if it will never halt or you haven't waited long enough for it to terminate.