algorithms

steps that a computer program may follow to accomplish a task

Types of algorithms

Routing algorithms

Decide on "best" path to reaching a goal

Graph algorithms

Sorting

Search

Recursive

Value of knowing algorithms

Take advantage of already known algorithms

What are examples of tasks which algorithms are designed to solve?

Metrics of an algorithm?

Efficiency?

Does it solve the problem at hand? Is it "correct"?

Asmytotic analysis

Analysis of worst-case scenerios

Focusses solely on the algorithm, and removes the effect of variables such as processor speed, etc

Analyzes an algorithm as a function of the size of its inputs

Analyzes the rate at which runtime increases as inputs increase

Runtime is measured as the number of primitive operations which are required to achieve the desired solution as an effect of inputs

Focuses on the those aspects of an algorithm which most affect the growth rate of runtime as inputs increase

Sort

Selection sort

Insertion sort

Merge sort

Quick sort

Search

Linear search

Very inefficiant

Binary search

a halving algorithm

Breadth first search

pseudocode

A way of describing an algorithm including only the details that need to be communicated for comprehension

Leaves out details not important to describing the algorithm

why pseudocode?

provides a succinct, efficient language for communicating a problem

General

Algorithms are strongly related to data structures

# Concepts

Divide and conquer