# Algorithms

## Introduction
## (Based on [Manber 1989])

Yih-Kuen Tsay

Department of Information Management
National Taiwan University

## What They Are

- An **algorithm** is, broadly speaking, a *step-by-step* procedure for solving a problem or accomplishing some end.
- When it is meant for the computer, each step in an algorithm should be realizable by *well-defined*, limited *primitive* operations that the computer understands.
- Algorithm design is an important and usually the hardest part of programming (which consists in finding/devising a solution and translating it into a computer program).
- Better algorithms (designed once, used forever) save more time and money.

## Development of an Algorithm

- We typically are given a problem statement, including input and output requirements, that is an abstract yet *accurate* and *precise* account of the problem to be solved and the properties of a satisfactory solution.
- The development of an algorithm involves the following tasks:
  1. Design (main subject of this course)
  2. Verification (or Proof of Correctness)
  3. Analysis
  4. Implementation

  (May need to iterate.)

## Main Concerns

- Why is algorithm design difficult?
  - Computers are different from humans; they are very fast and can handle much larger amounts of data.
  - Counterintuitive approaches may be needed, because of large problem scales.
  - Better solutions, if worthwhile (with greater payoffs), may be more complicated.
- How do we approach it?

## Creative Approach to the Subject

- Emphasis of the creative side
  - not only memorizing solutions
  - but also learning to create by trying to create
- Induction as one central design method
  - to explain/understand the principles behind a design
  - to systematically guide the creation process

## Design by Induction

- Design by induction draws analogies from proving theorems by *mathematical induction*.
- In a proof by induction, we do not prove a statement from scratch, but rather we show
    1. the correctness of the statement follows from that of the same statement for smaller instances and
    2. the correctness of the statement for a small base case.
- This suggests an approach to algorithm design that concentrates on *extending* solutions for smaller problem instances to solutions for larger ones.
- Induction may not solve every problem, but is very helpful.