Session 30: Algorithms

- Introduction to Algorithms

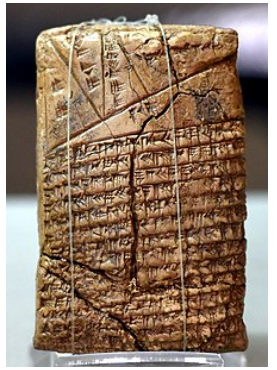
What is an algorithm?

Finite set of **well-defined** instructions to perform a specified **task**

- to perform a computation
- to solve a certain problem
- to reach a certain destination

In the context of computing the instructions are implemented by a computer

History of Algorithms



- Performing a division (Babylon, 2500BC)
- Finding prime numbers (Eratosthenes, 300BC)
- Solving quadratic equations (Muhammad ibn Musa al-Khwarizmi, 780AD) - al-Khwarizmi ~ algorithm/algebra
- Breaking encrypted messages (Al-Kindi 800AD)
- Deciding propositional logic (Hilbert, Gödel, Church, Kleene, Ross, Post, 20th century)
- Turing machine - algorithms on a computer

Alan Turing, 1912 - 1954

Example

Task: Find the maximum value in a finite sequence of integers.

Algorithm:

1. Set the temporary maximum equal to the first integer in the sequence.
2. Compare the next integer in the sequence to the temporary maximum. If it is larger than the temporary maximum, set the temporary maximum equal to this integer.
3. Repeat the previous step if there are more integers. If not, stop.
4. When the algorithm terminates, the temporary maximum is the largest integer in the sequence.

Example

Sequence	3	5	1	7	2	1
Temporary maximum						

Specifying Algorithms

- Algorithms can be specified in different ways.
  - Natural language
  - Pseudo-code
  - Programming language
- Pseudocode is an intermediate step between a Natural Language description (more precise) and a coding of these steps using a programming language (more general).
  - Programmers can use the description of an algorithm in pseudocode to construct a program in a particular language.
  - Pseudocode helps us analyse the properties of an algorithm, independent of the actual programming language used to implement it.

Example

Task: Find the maximum value in a finite sequence of integers.

Algorithm in pseudocode:

```
procedure max(a_1, a_2, …., a_n: integers)
   tmp_max := a_1
   for i := 2 to n
       if tmp_max < a_i then tmp_max := a_i
   return tmp_max
```

Example

Task: Find the maximum value in a finite sequence of integers.

Algorithm in Python:

```
def max(a):
    tmp_max = a[0]
    for i in range(2, len(a)):
        if tmp_max < a[i]:
            tmp_max = a[i]
    return tmp_max
```

```
max([2,5,3,7,4,1])
```

```
7
```

Typical Problems Solved by Algorithms

1. *Searching problems*: finding the position of a particular element in a list.
2. *Sorting problems*: putting the elements of a list into increasing order.
3. *Optimization Problems*: determining the optimal value (maximum or minimum) of a particular quantity over all possible inputs.

Summary

- Definition of Algorithm
- Pseudocode
- Types of Algorithms