



# Algorithms

## Algorithm Classification

Paolo Camurati and Stefano Quer

Dipartimento di Automatica e Informatica

Politecnico di Torino

# Algorithms

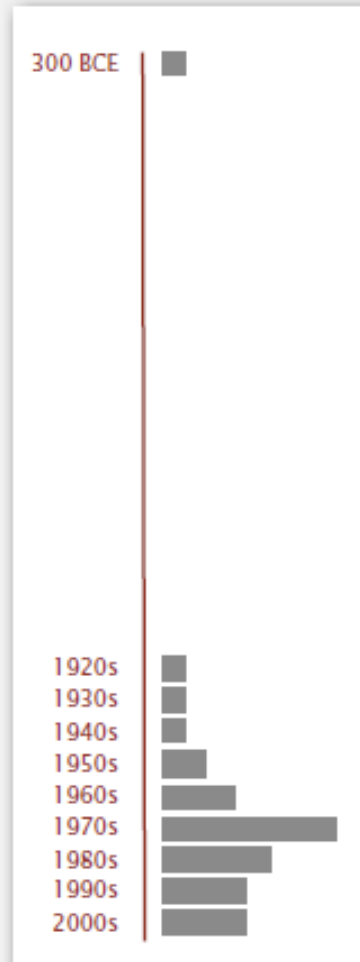
## ❖ Algorithms

- Finite sequence of instructions that solve a problem
- Satisfy the following criteria
  - They receive input values
  - They produce output values
  - They are clear, unambiguous and executable
  - They terminate after a finite number of steps
  - They work on data structures

"Programming is learned by writing programs"  
Brian Kernighan

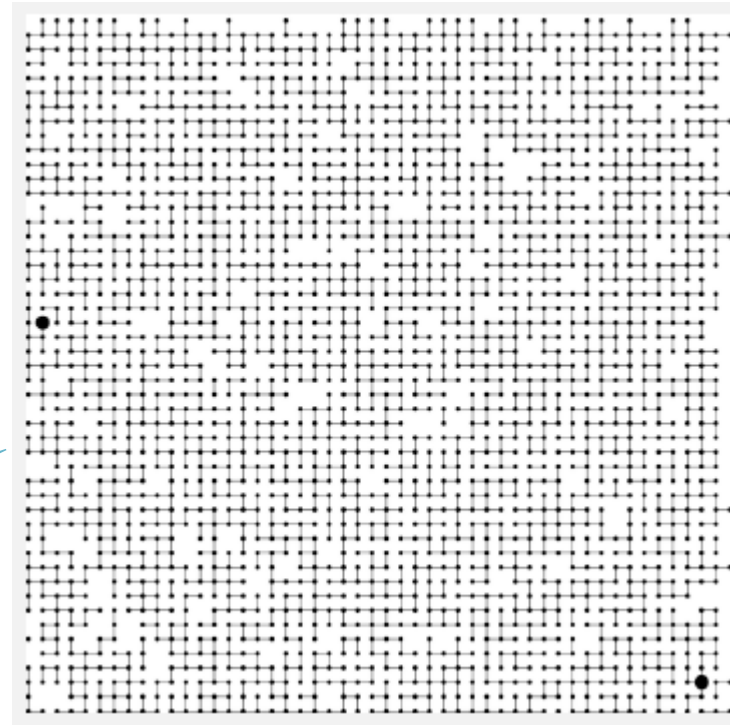
## Why study algorithms?

- ❖ Algorithms have ancient roots
  - Euclid, Greek, IV cent, B.C.
  - Al-Huarizmi, Persian IX cent., A.D.
  - Formalization by Church and Turing (1930s)
  - Recent developments



## Why study algorithms?

- ❖ To do something otherwise impossible
  - Are the two dark dots connected in this network (network connectivity)?
  - ... yes ...



Network connectivity

## Why study algorithms?

- ❖ Their impact is broad and far-reaching
  - Internet: Web search, packet routing, distributed file sharing
  - Biology: human genome
  - Computers: CAD tools, file systems, compilers
  - Graphics: virtual reality, videographics
  - Multimedia: MP3, JPG, DivX, HDTV
  - Social Networks: recommendations, news feed, advertisement
  - Security: e-commerce, cell phones
  - Physics: particle collision simulation

## Why study algorithms?

- ❖ To unlock the secrets of life and of the universe by creating models
  - In many sciences computational models are replacing mathematical ones

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left( a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right)$$
$$\sin \alpha \pm \sin \beta = 2 \sin \frac{1}{2}(\alpha \pm \beta) \cos \frac{1}{2}(\alpha \mp \beta)$$

20<sup>th</sup> century science  
Mathematical formulae  
(formula based)

```
int f (int **m, int n, int m) {  
    int i, j, sum;  
    sum = 0;  
    for (i=0; i<n; i++) {  
        for (j=0; j<m; j++) {  
            sum+=m[i][j];  
        }  
    }  
    return sum;  
}
```

21<sup>th</sup> century science  
Computational model  
(algorithm based)

## Why study algorithms?

### ❖ For intellectual stimulation

- "An algorithm must be seen to be believed."

Donald Knuth

### ❖ To become a proficient programmer

- "I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships."

Linus Torvalds (creator of Linux)

### ❖ For fun, for money, to increase speed, to process more data, ...

## Types of problems

### ❖ Decision problems

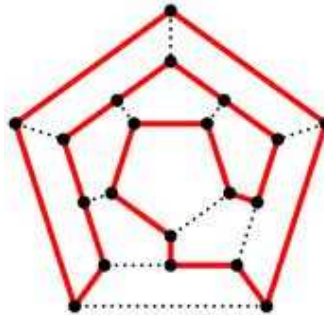
- Problems with a yes/no answer
- They may be
  - Decidable
    - There exists an algorithm that solves them
  - Undecidable
    - There is no algorithm that solves them
- Examples
  - Given 2 integers  $x$  and  $y$ , does  $x$  exactly divide  $y$ ?
  - Given a positive integer  $x$ , is it prime?
  - Given a positive integer  $n$ , do 2 positive and  $> 1$  integers  $p$  and  $q$  exist such that  $n = pq$ ?
  - Determine whether a number is prime



## Types of problems

### ❖ Search problem

- Does a valid solution exist and which one is it?



### ➤ Examples

- Hamiltonian cycle
  - Given an undirected graph, does a simple cycle spanning all vertices exist? Which one is it?
- Which one is the k-th prime number
- Given an array of integers, sort it in ascending order



## Types of problems

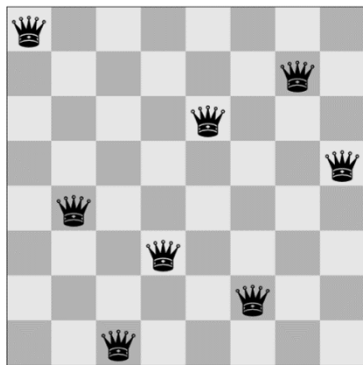
### ❖ Verification problems

➤ Given a solution (certificate), make sure that it is really one

➤ Examples

- Sudoku
- The eight queen problem

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9



## Types of problems

### ❖ Optimization problems

- If a solution exists, which one is the best one?
- Examples
  - Given a weighted directed graph, which is the shortest simple path, if it exists, between nodes  $i$  and  $j$ ?

