Informatica @ UniUrb: Offerta Formativa 3 + 2

Prof. Marco Bernardo

Università degli Studi di Urbino Carlo Bo
Dipartimento di Scienze Pure e Applicate

Scuola di Scienze, Tecnologie e Filosofia dell'Informazione

Corso di Laurea Triennale in Informatica – Scienza e Tecnologia
Corsi di Laurea Magistrale in Informatica e Innovazione Digitale
`https://informatica.uniurb.it/`

https://informatica.uniurb.it/

Cosa Intendiamo per Informatica?

- Informatics: discipline that studies automatic information processing.
- People perceive only technological aspects, while informaticians have to be aware of scientific and methodological aspects too.
- Scientific aspects – *Computer Science*: computability theory, complexity theory, algorithmics, automata theory, formal languages.
- Methodological aspects – *Software Architecture & Engineering*: programming methodologies, languages, and environments, etc.
- Technological aspects – *Information & Communication Technology*: computers, operating systems, data bases, computer networks, etc.

- Computer: set of programmable electromechanical devices for inputting, storing, processing, and outputting information in the form of numbers, texts, images, audios, and videos.
- There are many sets of electromechanical devices that can execute *several* functions (lifts, household appliances, vehicles, etc.) but only computers can execute a potentially *unlimited* number of functions, i.e., only computers are programmable machines!
- Hardware: set of electromechanical devices that form a computer, i.e., computational resources that are made available (*physical parts*).
- Software: set of programs that can run on a computer, i.e., instructions given to its computational resources to carry out tasks (*immaterial parts* – name originated from Jacquard looms).

- Socio-economic impact of informatics since 1950 arising from:
  - Transferral of repetitive or complex activities from people to machines.
  - Execution of complicated calculations in a short time.
  - Process/retrieve/transmit large amounts of information in a short time.
- The invention of printing in 1400 started to foster in the world a higher diffusion of knowledge.
- The industrial revolution in 1700 widened our physical capabilities through the introduction of automatic machines.
- The digital transformation based on electronics and informatics is increasingly extending our cognitive capabilities through programmable devices like computers and smartphones and the propagation of digital data over Internet.

- Computational thinking is the cultural contribution of informatics, intended as a mental process aiming at problem solving.
- Expression coined by Jeannette Wing (2006).
- Combination of:
  - Characteristic methods of informatics:
    - Algorithmic analysis of problems.
    - Digital representation of data.
    - Automation of solutions.
  - General intellectual capabilities:
    - Facing complexity.
    - Comparing alternatives.
- Computational variant of abstract thinking typical of mathematics: *software is immaterial!*

- Given a computational problem, the theory of computation studies the existence of an algorithmic solution and, if any, the needed quantity of computational resources in terms of:
  - Running time.
  - Memory space.
  - Communication bandwidth.
- Computability theory: decidable vs. undecidable problems depending on the existence of an algorithmic solution or not.
- Complexity theory: tractable vs. intractable problems depending on polynomially or exponentially many resources.
- If a computational problem is decidable, there might be several different algorithms solving it more or less efficiently (insertsort, selectsort, bubblesort, quicksort, mergesort, heapsort).

- Algorithm: finite sequence of steps represented in such a way that they are intelligible by an executor to solve a computational problem in its generality (i.e., for all instances of its input data).
- Persian mathematician Muhammad ibn Musa al-Khwarizmi (780–850) is the author of one of the most ancient treatises of algebra (al-jabr) in which he described how to solve linear and quadratic equations.
- Although the representation of an algorithm has a finite length, the duration of its execution can be unbounded (e.g., loops).
- The executor is not necessarily an electromechanical agent, can be a biological agent or a cyberphysical agent (e.g., recipes).
- Program: algorithm expressed in a specific programming language, whose executor will be a computer (software, application, code, ...).

- Information and communication technology is *pervasive*.
- Malfunctioning, poor performance, security breaches, bad interfaces.
- These errors result in waste of time and money for software producers.
- Human losses and environmental damages if software is safety-critical!
- `https://en.wikipedia.org/wiki/List_of_software_bugs`
- Methodologies for guiding software design, development, deployment.
- Software testing does not guarantee the absence of errors.
- Software verification is needed:
  - Computational model of the program.
  - Formalization of the properties of interest.
  - Generation of counterexamples to explain property violations.

Cenni di Storia dell'Informatica

- Pioneers between 1600 and 1800 (mechanical technology):
  - Blaise Pascal (1623–1662):
    - Mechanical machine capable to do additions and subtractions (1642).
  - Gottfried Wilhelm von Leibniz (1646–1716):
    - Mechanical machine including multiplications and divisions too (1672).
    - Binary system, characteristica universalis, calculus ratiocinator.
  - Charles Babbage (1791–1871):
    - Difference Engine (1822): tabulation of polynomial functions and hence approximation of logarithmic and trigonometric functions.
    - Analytical Engine (1834): archetype of all modern computers both for its architecture and for its instruction set.
  - Ada Byron Lovelace (1815–1852):
    - Translation of and notes on the Analytical Engine project (1843).
    - Bernoulli numbers: *she was the first programmer of the history!*

- Some pioneers of 1900 (electronic technology):
  - Norbert Wiener (1894–1964):
    - Cybernetics: interdisciplinary study of natural and artificial systems in terms of control, feedback, and communication.
  - Claude Shannon (1916–2001):
    - Connection between electrical circuits and George Boole algebra.
    - Information theory: study of the encoding and the transmission of digital information, whose elementary unit he called bit.
  - Alan Turing (1912–1954):
    - Turing machine (TM): first formalization of the concept of algorithm.
    - Universal Turing machine (UTM): vision of software in terms of computation scheme no longer hardwired.
    - Cryptography and artificial intelligence.
  - John Von Neumann (1903–1957):
    - From plugboard computers to stored program computers (UTMs).
    - From serial decimal arithmetic to parallel binary arithmetic.
    - Game theory.

- Vacuum tubes → transistors → integrated circuits → VLSI → ...
- Increasing computation speeds and storage capacities.
- Decreasing costs and sizes (desktop/laptop personal computers).
- Italian pioneers:
  - Adriano Olivetti (1901–1960):
    - Elea 9003 (1957): *first computer entirely based on transistors*, designed at Olivetti by Mario Tchou (1924–1961).
    - P101 (1964): *first personal computer of the history*, designed at Olivetti by Pier Giorgio Perotto (1930–2002).
  - CEP – Calcolatrice Elettronica Pisana (1961):
    - First scientific computer designed in Italy, at the University of Pisa, an initiative promoted by Enrico Fermi and Adriano Olivetti.
  - Federico Faggin (1941):
    - Designed the first microprocessor of the history at Intel (1971).
    - Developed the first touchpads and touchscreens (late 1980's).

https://informatica.uniurb.it/

Genesi di Informatica a UniUrb

- Il corso di laurea triennale in Informatica Applicata di UniUrb venne attivato nell'a.a. 2001/2002 come "progetto di Ateneo con il coinvolgimento e la partecipazione di diverse facoltà".
- Il Rettore Carlo Bo nominò nel 2000 una commissione incaricata di:
  - "fornire indicazioni sulla organizzazione delle attività didattiche";
  - "suggerire i raccordi fondamentali con realtà produttive e di ricerca".
- Presieduta dal Preside Mauro Magnani e composta da docenti di UniBo e PoliTo, oltre che da Umberto Paolucci di Microsoft.
- Il corso di laurea magistrale fu poi attivato nell'a.a. 2020/2021.
- L'aggettivo Applicata denotava la sinergia tra più ambiti:
  - Scienze informatiche (aspetti teorici e metodologici).
  - Ingegneria dell'informazione (aspetti sistemistici e applicativi).
  - Ingegneria elettronica.
- Nuove epigrafi e immagine dall'a.a. 2023/2024.

Piano degli Studi della Triennale

- Primo anno:
  - Programmazione Procedurale.
  - Algoritmi e Strutture Dati.
  - Reti Logiche.
  - Architettura degli Elaboratori.
- Secondo anno:
  - Programmazione e Modellazione a Oggetti.
  - Ingegneria e Architettura del Software.
  - Sistemi Operativi.
- Terzo anno:
  - Programmazione Logica e Funzionale.
  - Linguaggi di Programmazione e Verifica del Software.
  - Basi di Dati.
  - Reti di Calcolatori.

- Pilastro programmazione software + pilastro sistemi di elaborazione.
- Linguaggi di programmazione insegnati nel corso dei tre anni:
  - C $\rightsquigarrow$ linguaggio imperativo procedurale.
  - Java, C++, C#, Python $\rightsquigarrow$ linguaggi imperativi a oggetti.
  - Haskell $\rightsquigarrow$ linguaggio dichiarativo funzionale.
  - Prolog $\rightsquigarrow$ linguaggio dichiarativo logico.
- Altri linguaggi insegnati:
  - UML $\rightsquigarrow$ linguaggio di modellazione di software a oggetti.
  - SQL $\rightsquigarrow$ linguaggio di interrogazione di basi di dati.
- Ogni insegnamento di area informatica si compone di due parti:
  - Una parte teorica, in aule dotate di computer con videoproiettore, lavagna di ardesia con webcam fissa, touch board con smart webcam.
  - Una parte pratica, in un laboratorio dotato di personal computer coi sistemi operativi Linux (open source) e Windows (proprietario).
- L'esame comprende un progetto, una prova scritta e una prova orale.

- Formazione matematico-fisica:
  - Logica, Algebra e Geometria (I).
  - Analisi Matematica 1 (I).
  - Analisi Matematica 2 (II).
  - Probabilità e Statistica Matematica (II).
  - Fisica Generale (II).
- Curricula tra cui scegliere (II, III):
  - Segnali, Simulazione e Quantum Computing.
  - Piattaforme Digitali per il Monitoraggio del Territorio.
  - Economia e Diritto dell'Informatica.
  - Comunicazione e Media Digitali.
- Insegnamenti erogati come MOOC:
  - Pensiero Computazionale in Classe (I).
  - Umano Digitale (I).
- Insegnamenti a scelta, seminari, tirocini/stage, tesi.

All'Inizio del Percorso Triennale

- L'immatricolazione avviene in modalità telematica.
- L'iscrizione alla triennale prevede il sostenimento di una prova di verifica della preparazione iniziale (VPI) attraverso TOLC-S o TOLC-I.
- In settembre vengono organizzati dei precorsi di matematica sia per preparare alla prova di VPI che per colmare eventuali lacune.
- Se la prova non viene superata o sostenuta nei tempi previsti, vengono attribuiti degli obblighi formativi aggiuntivi (OFA).
- Chi ha OFA da assolvere non può sostenere gli esami di insegnamenti del secondo e terzo anno!
- Il percorso triennale prevede l'acquisizione di 180 CFU, dove 1 CFU = 25 ore di attività da parte di chi studia (frequenza lezioni, studio individuale, tirocinio/stage, tesi).

Durante il Percorso Triennale

- Le lezioni degli insegnamenti di area informatica del primo anno non presuppongono il possesso di conoscenze pregresse ...
- ... ma è fondamentale studiare con regolarità!
- La frequenza a lezioni ed esercitazioni è tuttavia consigliata.
- Piano di studi diluito su 6 anni per chi lavora.
- Due periodi didattici:
  - Primo semestre da fine settembre a fine dicembre (12 settimane).
  - Secondo semestre da fine febbraio a fine maggio (12 settimane).
- Tre sessioni d'esame:
  - Sessione invernale da metà gennaio a fine febbraio (6 settimane).
  - Sessione estiva da inizio giugno a metà luglio (6 settimane).
  - Sessione autunnale da fine agosto a fine settembre (4 settimane).
- Seminari tenuti da esperti e stage presso aziende.

Alla Fine del Percorso Triennale

- La laurea in Informatica – Scienza e Tecnologia consente l'accesso:
  - alle lauree magistrali in Informatica col riconoscimento dei 180 CFU;
  - all'Esame di Stato per l'iscrizione all'Albo degli Ingegneri Juniores;
  - al mondo del lavoro con ottime prospettive (fonte AlmaLaurea).
- Profili professionali:
  - Tecnico programmatore.
  - Tecnico esperto in applicazioni.
  - Tecnico web.
  - Tecnico gestore di basi di dati.
  - Tecnico gestore di reti e di sistemi telematici.
- Sbocchi occupazionali:
  - Libera professione nel settore ICT.
  - Società di consulenza, progettazione, fornitura servizi digitali.
  - Aziende e organizzazioni pubbliche e private con esigenze informatiche.

Percorso Biennale della Laurea Magistrale

- Primo anno:
  - Programmazione di Dispositivi Mobili e Interfacce Utente.
  - Sistemi Distribuiti.
  - Sicurezza Informatica.
  - Machine Learning.
- Secondo anno:
  - Programmazione per l'Internet of Things.
  - Applicazioni Distribuite e Cloud Computing.
- Formazione matematico-fisica:
  - Metodi Numerici per l'Algebra Lineare e l'Analisi Funzionale (I).
  - Elaborazione dei Dati Sperimentali (II).
- Curricula tra cui scegliere:
  - Intelligenza Artificiale.
  - Analisi Statistico-Economica per le Imprese.
  - Analisi Sociologica delle Tecnologie Digitali.

- Direzioni strategiche ed esigenze socio-economiche:
  - Digitalizzazione e dematerializzazione.
  - Crescita del mercato digitale.
  - Industria e impresa 4.0.
  - AI per la green economy.
  - App per smart device.
- Didattica erogata in modalità mista presenza-distanza
  per venire incontro alle esigenze di chi già lavora.
- La laurea in Informatica e Innovazione Digitale consente l'accesso:
  - ai dottorati di ricerca in Informatica;
  - all'Esame di Stato per l'iscrizione all'Albo degli Ingegneri Seniores;
  - al mondo del lavoro con ottime prospettive (fonte AlmaLaurea).
- Profili professionali:
  - Mobile/IoT system and application designer/developer.
  - Security/Network specialist.
  - Big data analyst.

Servizi e Opportunità di UniUrb

- Identità digitale per l'utilizzo di tutti i servizi telematici.
- Posta elettronica, cloud storage e webconference (Google suite).
- WiFi disponibile in tutti gli edifici dell'Ateneo.
- Gestione dematerializzata della carriera studentesca.
- Piattaforma Moodle per materiale didattico, prove di autovalutazione e forum di discussione docenti-studenti (blended learning).
- Competenze trasversali attraverso corsi erogati dal CISDEL – Centro Integrato Servizi Didattici ed E-Learning di UniUrb.
- Servizi sportivi, di tutorato e di supporto psicologico.
- Servizi di ristorazione e alloggio gestiti da ERDiS.
- Borse di studio e agevolazioni tasse di iscrizione.

https://informatica.uniurb.it/

Programmi Ricorsivi e Principio di Induzione

- Un problema complesso viene suddiviso in sottoproblemi più semplici così la soluzione viene ottenuta combinando quelle dei sottoproblemi.
- Quando i sottoproblemi sono della *stessa natura del problema originale* si può adottare uno schema di soluzione ricorsivo:
  - Individuare uno o più casi base, per i quali si può ricavare direttamente la soluzione del problema.
  - Definire uno o più casi generali attraverso un insieme di sottoproblemi della stessa natura di quello originale ma più vicini ai casi base.
- La ricorsione è uno strumento molto potente:
  - Affrontare problemi altrimenti ingestibili (problema delle torri di Hanoi).
  - Risolvere problemi in maniera più efficiente (mergesort e quicksort).
  - Definire in modo naturale certe strutture dati (liste e alberi).
- La sua base teorica è il principio di induzione.

- $\mathbb{N}$ è infinito e quindi non può essere definito elencandone gli elementi.
- Definizione assiomatica di Dedekind-Peano in teoria degli insiemi:
  1. Esiste un elemento $0 \in \mathbb{N}$.
  2. Esiste una funzione totale $succ : \mathbb{N} \to \mathbb{N}$.
  3. Per ogni $n \in \mathbb{N}$, $succ(n) \neq 0$.
  4. Per ogni $n, n' \in \mathbb{N}$, se $n \neq n'$ allora $succ(n) \neq succ(n')$.
  5. Se M è un sottoinsieme di $\mathbb{N}$ tale che:
     - $0 \in M$;
     - per ogni $n \in \mathbb{N}$, $n \in M$ implica $succ(n) \in M$;

     allora $M = \mathbb{N}$ (quindi $\mathbb{N}$ è il più piccolo insieme chiuso su 0 e $succ$).
- Gli elementi di $\mathbb{N}$ sono dunque 0, $succ(0)$, $succ(succ(0))$, ... dove $succ(0)$ è denotato da 1, $succ(succ(0))$ è denotato da 2, e così via.
- La sequenza finita di simboli della defin. genera un insieme infinito.

- L'ultimo assioma è il principio di induzione, uno dei più potenti strumenti della matematica discreta per definizioni e dimostrazioni.
- Definizione formale dell'intera aritmetica: aritmetica di Peano.
- Sia $pred : \mathbb{N}_{\neq 0} \to \mathbb{N}$ tale che $pred(succ(n)) = n$ per ogni $n \in \mathbb{N}$ e $succ(pred(n)) = n$ per ogni $n \in \mathbb{N}_{\neq 0}$.
- Definizione formale dell'addizione:

$$m \oplus n = \begin{cases} m & \text{se } n = 0 \\ succ(m) \oplus pred(n) & \text{se } n \neq 0 \end{cases}$$

- Esempio: $5 \oplus 2 = 6 \oplus 1 = 7 \oplus 0 = 7$.
- Definizione formale delle relazioni d'ordine:
  - $m \leq n$ se e solo se esiste $m' \in \mathbb{N}$ tale che $m \oplus m' = n$.
  - $m < n$ se e solo se $m \leq n$ con $m \neq n$.
  - $m \geq n$ se e solo se $n \leq m$ ed $m > n$ se e solo se $n < m$.

- Definizione formale della sottrazione ($m \geq n$):

$$m \ominus n = \begin{cases} m & \text{se } n = 0 \\ pred(m) \ominus pred(n) & \text{se } n > 0 \end{cases}$$

- Definizione formale della moltiplicazione:

$$m \otimes n = \begin{cases} 0 & \text{se } n = 0 \\ m \oplus (m \otimes pred(n)) & \text{se } n > 0 \end{cases}$$

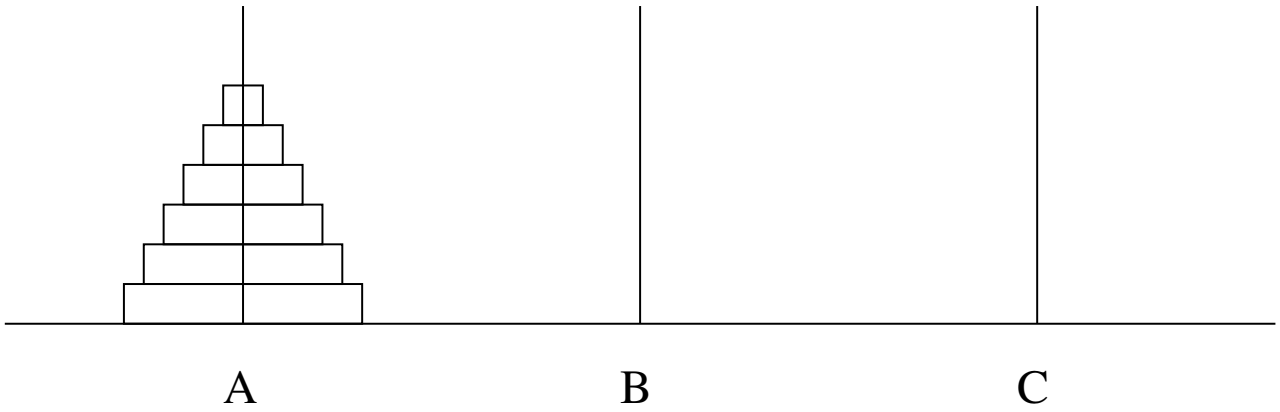
- Definizione formale della divisione ($n \neq 0$):

$$m \oslash n = \begin{cases} 0 & \text{se } m < n \\ succ((m \ominus n) \oslash n) & \text{se } m \geq n \end{cases}$$

- Esempi:
  - $5 \ominus 2 = 4 \ominus 1 = 3 \ominus 0 = 3$.
  - $5 \otimes 2 = 5 \oplus (5 \otimes 1) = 5 \oplus (5 \oplus (5 \otimes 0)) = 5 \oplus (5 \oplus 0) = 5 \oplus 5 = \ldots = 10$.
  - $5 \oslash 2 = succ((5 \ominus 2) \oslash 2) = \ldots = succ(3 \oslash 2) = succ(succ((3 \ominus 2) \oslash 2))$ $= \ldots = succ(succ(1 \oslash 2)) = succ(succ(0)) = succ(1) = 2$.

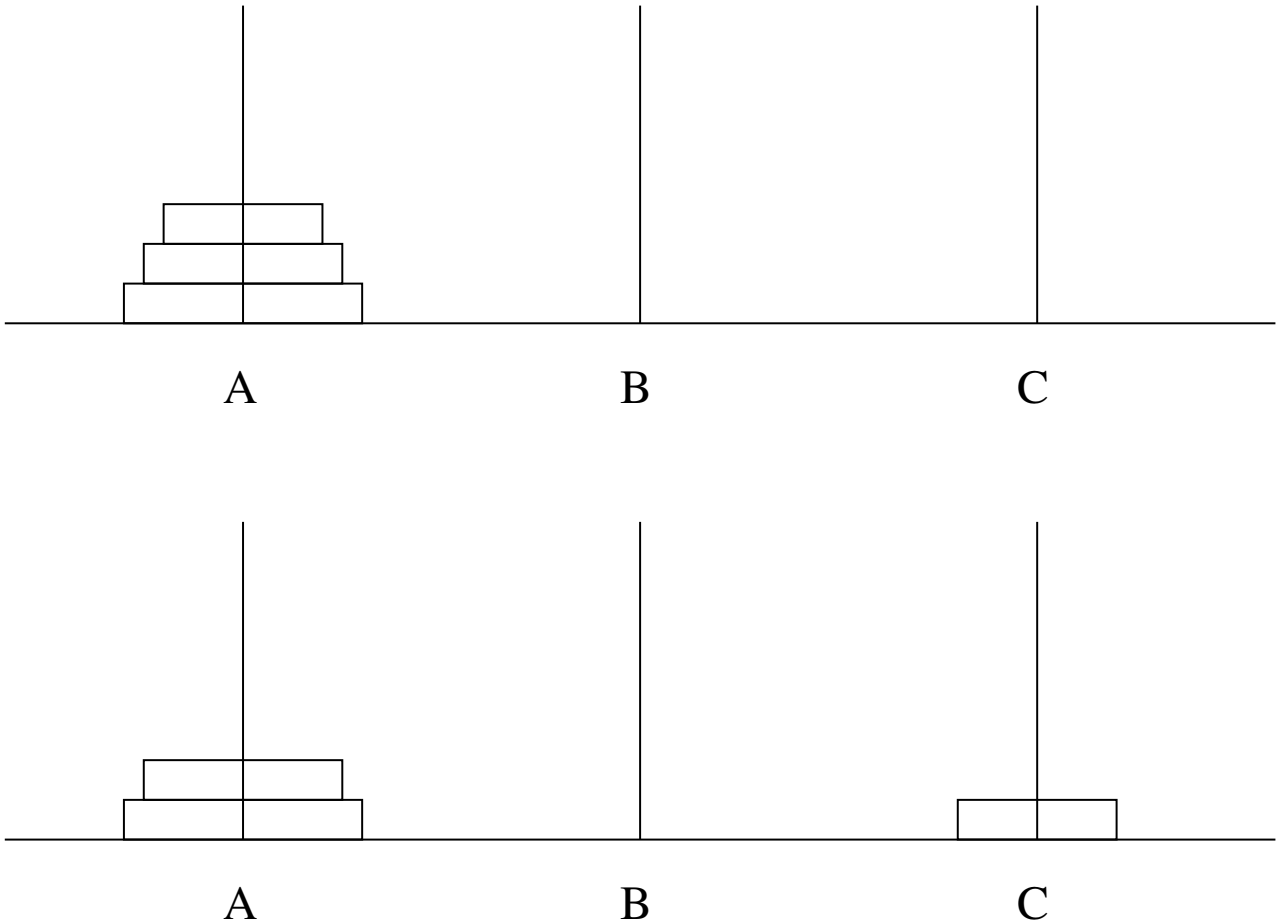
Il Problema delle Torri di Hanoi

- Siano date tre aste denominate A, B, C.
- Sull'asta A sono accatastati $n \geq 1$ dischi diversi tra loro in ordine di diametro decrescente dal basso verso l'alto:

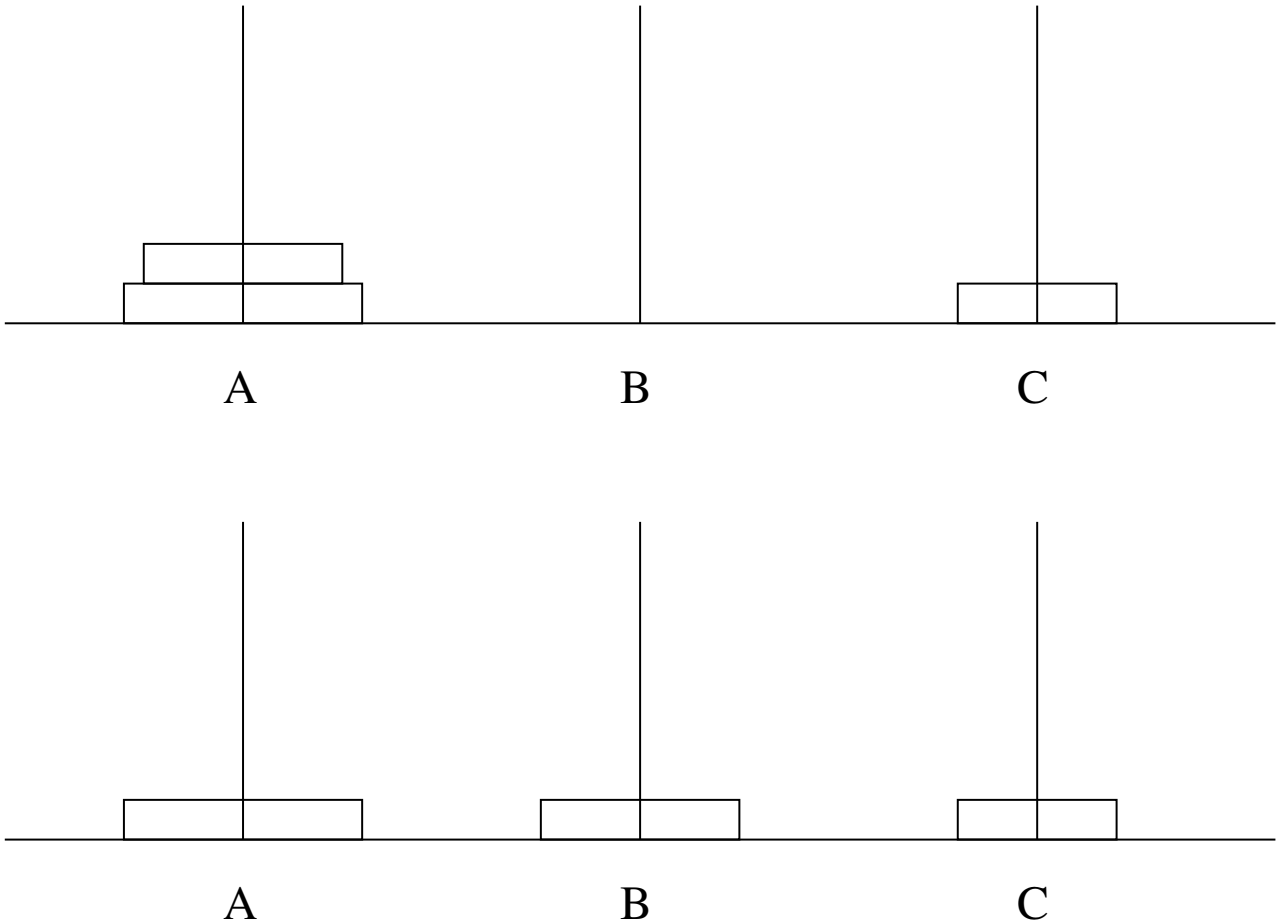


- Spostare i dischi da A a C rispettando le due seguenti regole:
  - È possibile spostare un solo disco alla volta.
  - Un disco non può mai trovarsi sopra uno di diametro inferiore.
- Soluzione banale quando $n = 1$ oppure $n = 2$.

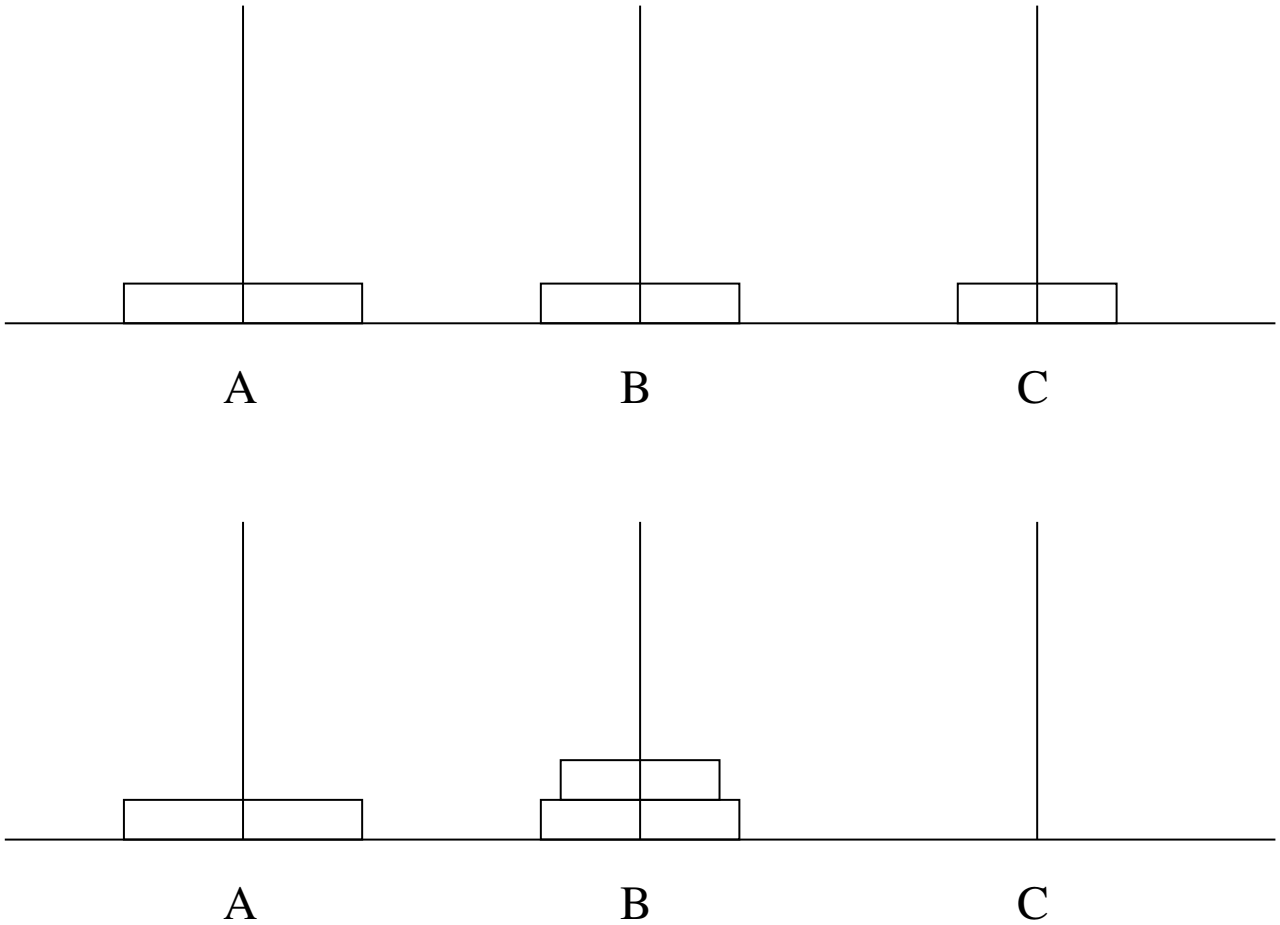
- Mossa 1 quando $n = 3$:



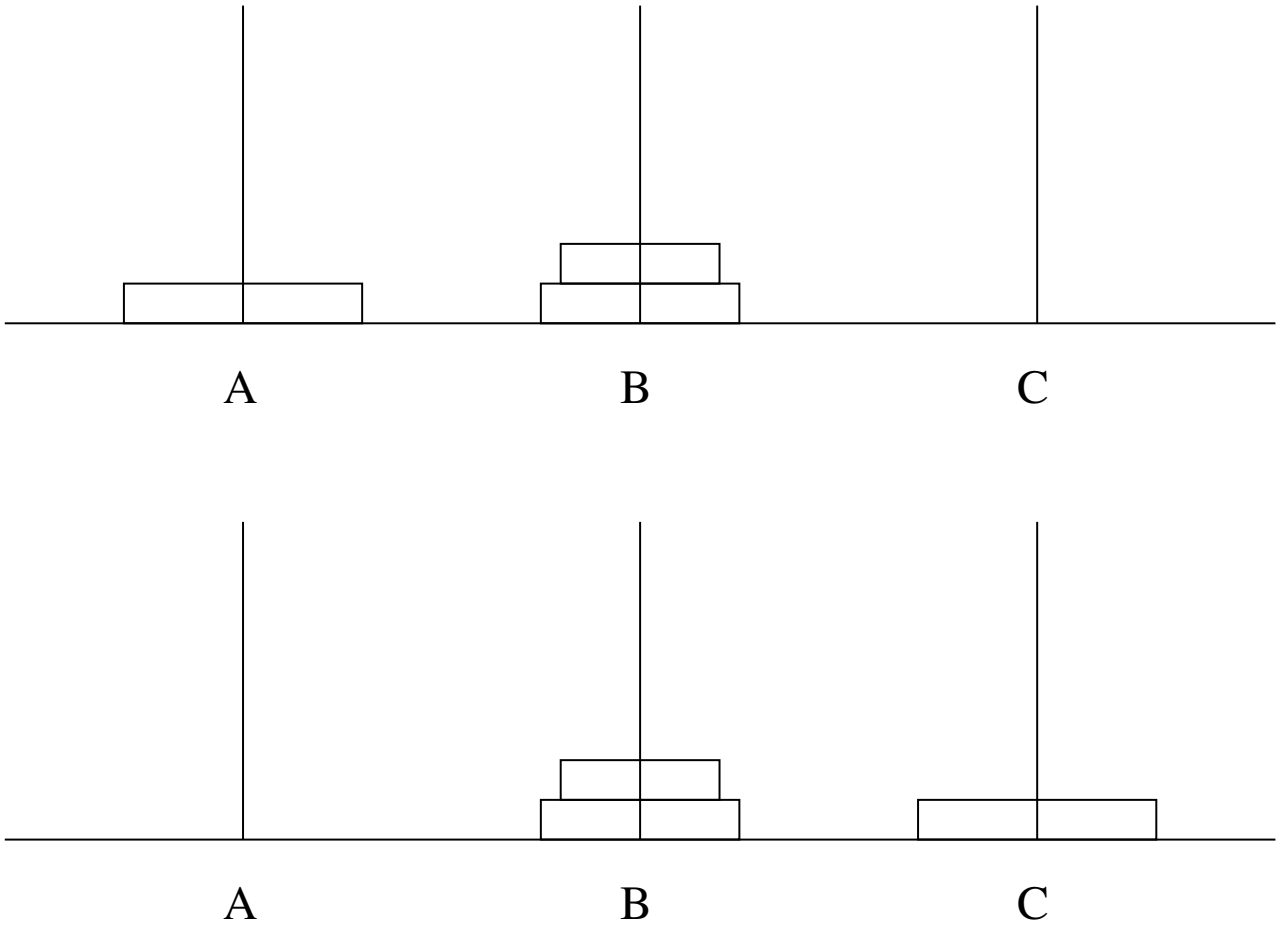
- Mossa 2 quando $n = 3$:



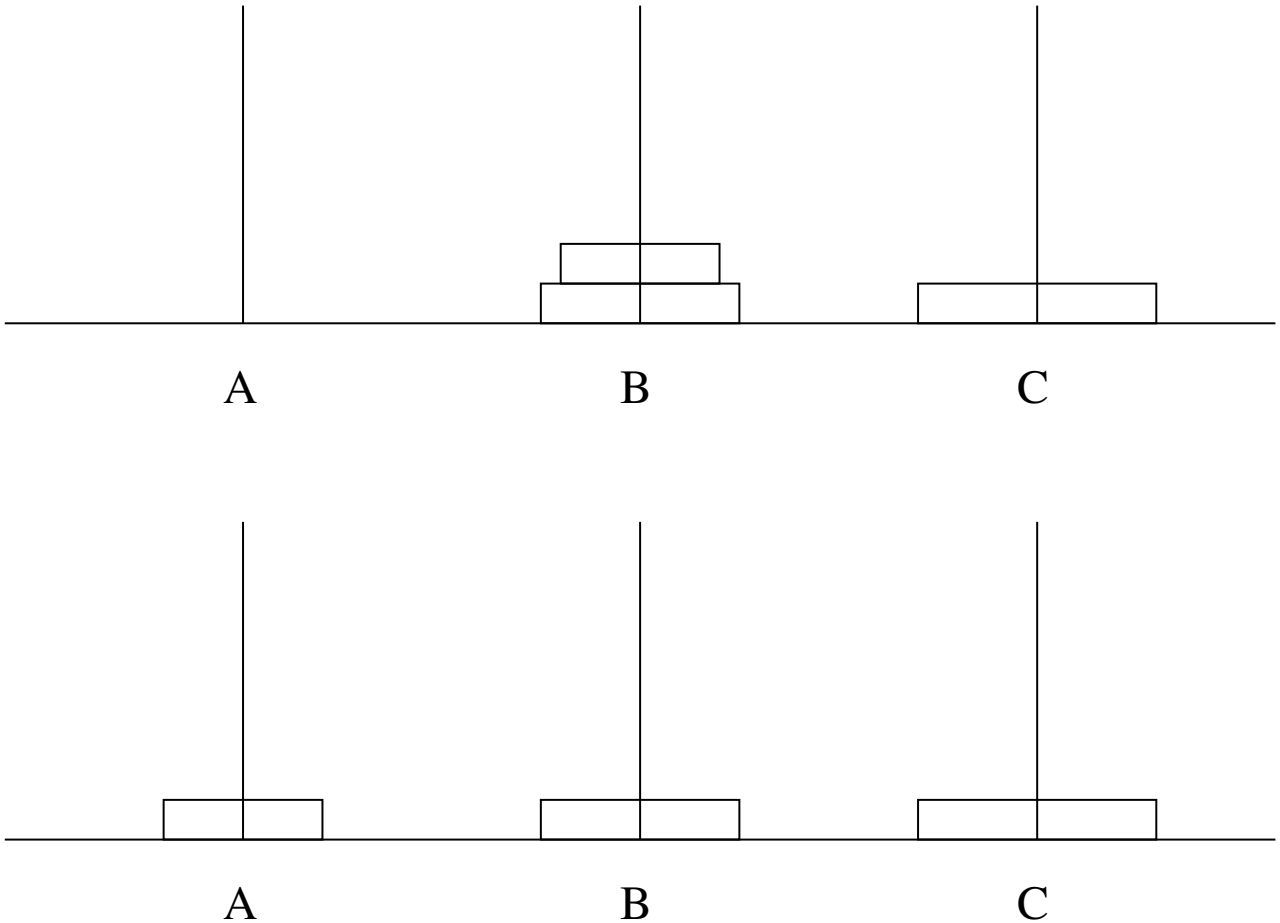
- Mossa 3 quando $n = 3$:



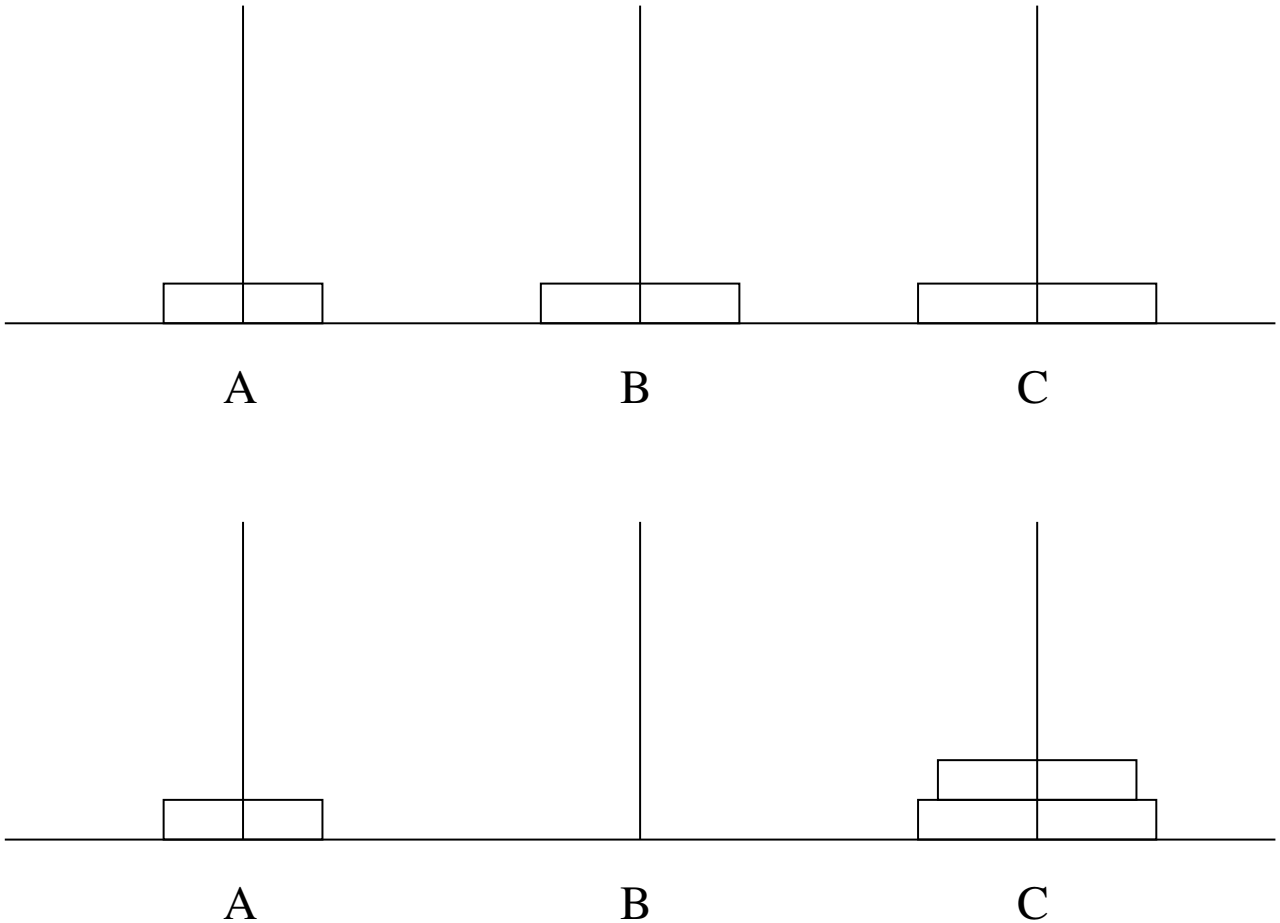
- Mossa 4 quando $n = 3$:



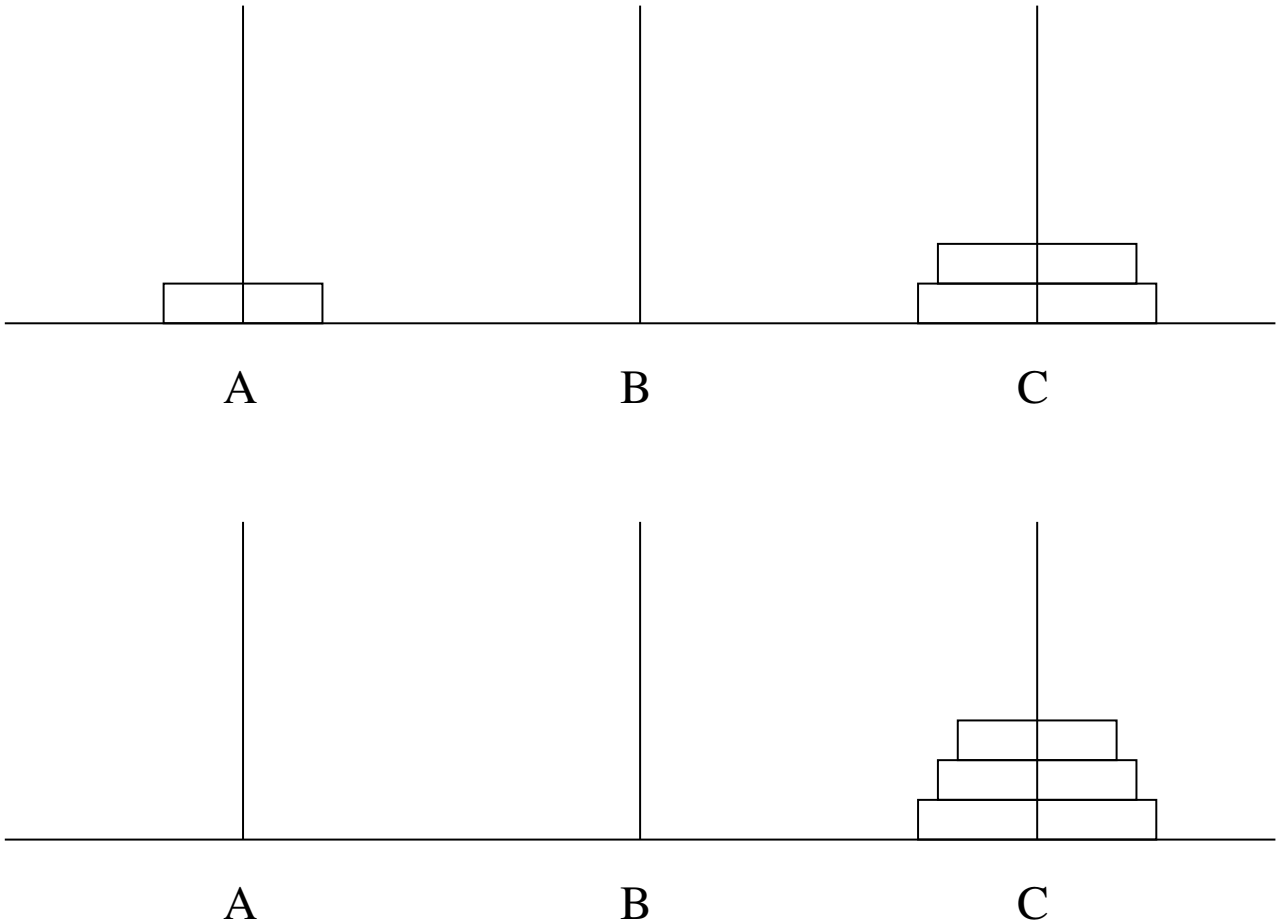
- Mossa 5 quando $n = 3$:



- Mossa 6 quando $n = 3$:



- Mossa 7 quando $n = 3$:



- Quando $n \geq 4$ diventa praticamente impossibile da risolvere a mano.
- Il numero di mosse aumenta esponenzialmente al crescere di n.
- Soluzione ricorsiva:
  - Se $n = 1$:
    - Spostare l'unico disco dalla prima alla terza asta.
  - Se $n > 1$:
    - Spostare gli $n - 1$ dischi più piccoli dalla prima alla seconda asta *usando questa soluzione ricorsiva*.
    - Spostare il disco più grande dalla prima alla terza asta.
    - Spostare gli $n - 1$ dischi più piccoli dalla seconda alla terza asta *usando questa soluzione ricorsiva*.
- Ad ogni passo ricorsivo cambia il ruolo delle tre aste.

- Implementazione nel linguaggio C:

```
void hanoi(int  n,              /* n >= 1 */
           char partenza,       /* 'A' nella chiamata iniziale */
           char arrivo,         /* 'C' nella chiamata iniziale */
           char intermedia)     /* 'B' nella chiamata iniziale */
{
  if (n == 1)
    printf("Sposta da %c a %c.\n",
           partenza,
           arrivo);
  else
  {
    hanoi(n - 1,
          partenza,
          intermedia,
          arrivo);
    printf("Sposta da %c a %c.\n",
           partenza,
           arrivo);
    hanoi(n - 1,
          intermedia,
          arrivo,
          partenza);
  }
}
```

https://informatica.uniurb.it/