# Trattabilità e Intrattabilità

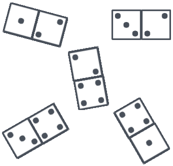
## Automi e Linguaggi Formali

LT in Informatica
a.a. 2018/2019

Università degli Studi di Padova

Disponete in fila le tessere del domino che vi sono state consegnate:

1. in modo da usare tutte le tessere

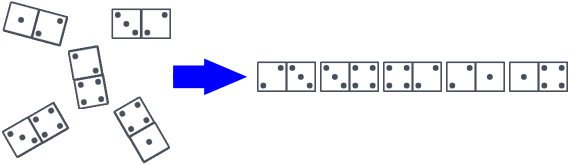


# Giochiamo a Domino[1]

Università degli Studi di Padova

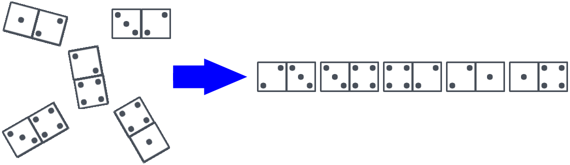
Disponete in fila le tessere del domino che vi sono state consegnate:

1. in modo da usare tutte le tessere



Disponete in fila le tessere del domino che vi sono state consegnate:

1. in modo da usare tutte le tessere



**Domanda:**

È un problema facile o difficile da risolvere?

# Come mostrare che un problema è facile?



## Definizione

Un problema è trattabile (facile) se esiste un algoritmo efficiente per risolverlo.

- Gli algoritmi efficienti sono algoritmi con complessità polinomiale:
  - il loro tempo di esecuzione è $O(n^k)$ per qualche costante $k$.
- Avere complessità polinomiale è una condizione minima per considerare un algoritmo efficiente
- Un algoritmo con complessità più che polinomiale (p.es. esponenziale) è un algoritmo non efficiente perché non è scalabile.

# Mostriamo che Domino[1] è trattabile



**Obiettivo**

Trovare un algoritmo polinomiale per Domino[1]

1. Formulazione del problema in termini di input e output trattabili da un calcolatore
2. Definizione dell'algoritmo
3. Analisi di complessità dell'algoritmo

# Dalle tessere al grafo



## Definition (Grafo)

Un grafo (non orientato) $G$ è una coppia $(V, E)$ dove:

- $V = \{v_1, v_2, \ldots, v_n\}$ è un insieme finito e non vuoto di vertici;
- $E \subseteq \{\{u, v\} \mid u, v \in V\}$ è un insieme di coppie non ordinate, ognuna delle quali corrisponde ad un arco del grafo.

# Dalle tessere al grafo

## Definition (Grafo)

Un grafo (non orientato) $G$ è una coppia $(V, E)$ dove:

- $V = \{v_1, v_2, \dots, v_n\}$ è un insieme finito e non vuoto di vertici;
- $E \subseteq \{\{u, v\} \mid u, v \in V\}$ è un insieme di coppie non ordinate, ognuna delle quali corrisponde ad un arco del grafo.

## Grafo del domino

- Vertici: i numeri che si trovano sulle tessere
  - $V =$ {⚀, ⚁, ⚂, ⚃}
- Archi: le tessere del domino
  - $E =$ {⚂⚃, ⚀⚁, ⚃⚁, ⚃⚀, ⚂⚁}

# Domino[1] è un problema su grafi!

Università degli Studi di Padova

**Domino[1]**: allineare tutte le tessere del domino

$\Longleftrightarrow$

**Cammino Euleriano**: trovare un percorso nel grafo che attraversa tutti gli archi una sola volta

- Il problema del cammino Euleriano è un problema classico di teoria dei grafi
- Esistono algoritmi polinomiali per risolverlo

# Algoritmo di Fleury



1. Scegliere un vertice con grado dispari (un vertice qualsiasi se tutti pari)
2. Scegliere un arco tale che sua cancellazione non sconnetta il grafo
3. Passare al vertice nell'altra estremità dell'arco scelto
4. Cancellare l'arco dal grafo
5. Ripetere i tre passi precedenti finche non eliminate tutti gli archi

**Complessità**

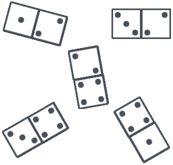
Su un grafo con $n$ archi, l'algoritmo di Fleury impiega tempo $O(n^2)$

# Giochiamo a domino[2]



Disponete in fila le tessere del domino che vi sono state consegnate:

2. in modo che ogni numero compaia esattamente due volte (potete usare meno tessere di quelle che avete).

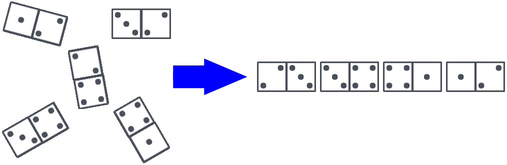


# Giochiamo a domino[2]



Disponete in fila le tessere del domino che vi sono state consegnate:

2 in modo che ogni numero compaia esattamente due volte (potete usare meno tessere di quelle che avete).

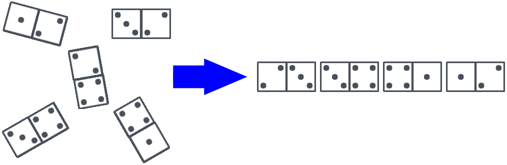


# Giochiamo a domino[2]



Disponete in fila le tessere del domino che vi sono state consegnate:

2 in modo che ogni numero compaia esattamente due volte (potete usare meno tessere di quelle che avete).



**Domanda:**

È un problema facile o difficile da risolvere?

# Domino[2] è un problema su grafi!



**Domino[2]**: trovare un allineamento dove ogni numero compare esattamente due volte

$\Longleftrightarrow$

**Circuito Hamiltoniano**: trovare un ciclo nel grafo che attraversa tutti i vertici una sola volta

- Il problema del circuito Hamiltoniano è un problema classico di teoria dei grafi
- Un algoritmo polinomiale per risolverlo non è mai stato trovato
- Se qualcuno mi dà una possibile soluzione, è facile verificare se è corretta

# Calendario



**Ultima settimana di lezione:**

- Oggi, lunedì 27 maggio 12:30–14:30
- Domani, martedì 28 maggio 12:30–14:30
- Venerdì 31 maggio 12:30–14:30

**Secondo compitino:**

- Venerdì 7 giugno, ore 12:30
- iscrizione su uniweb fino a mercoledì 5 giugno
- aperto a tutti
- su due aule: **LuM250** e **P200**
- distribuzione tra le aule: sul moodle, il giorno prima della prova

# Problemi trattabili e problemi intrattabili



- I problemi per i quali esiste un algoritmo polinomiale vengono considerati trattabili
- quelli che richiedono un algoritmo più che polinomiale sono detti intrattabili.
- Sappiamo che ci sono problemi che non possono essere risolti da nessun algoritmo:
  - "Halting Problem" di Turing
- Ci sono problemi che richiedono un tempo esponenziale:
  - il gioco della Torre di Hanoi

# Problemi trattabili e problemi intrattabili



- I problemi per i quali esiste un algoritmo polinomiale vengono considerati trattabili
- quelli che richiedono un algoritmo più che polinomiale sono detti intrattabili.
- Sappiamo che ci sono problemi che non possono essere risolti da nessun algoritmo:
  - "Halting Problem" di Turing
- Ci sono problemi che richiedono un tempo esponenziale:
  - il gioco della Torre di Hanoi

Stabilire con precisione qual'è il confine tra problemi trattabili ed intrattabili è piuttosto difficile

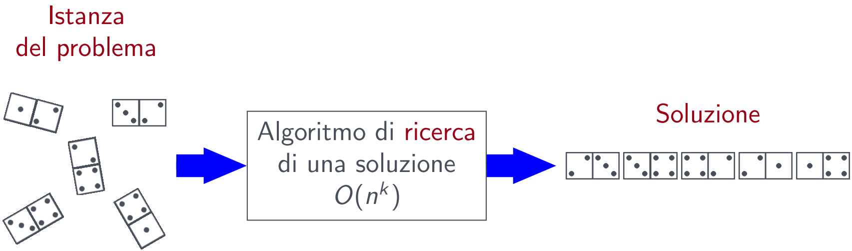
Facili da risolvere

Facili da verificare

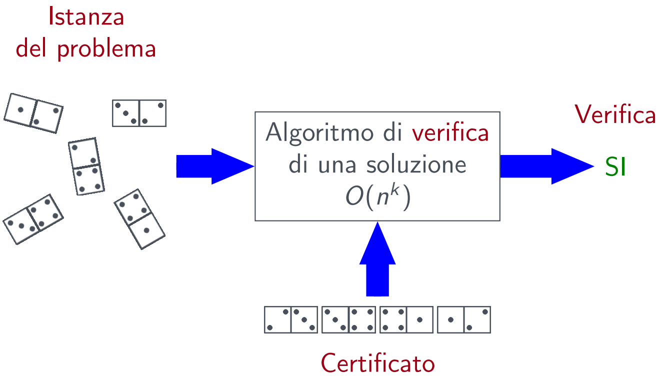
| | P |
|---|---|
| Facili da risolvere | ✓ |
| Facili da verificare | ✓ |
| Esempi | Domino[1], Euler, ordinamento, ... |

| | P | NP |
|---|---|---|
| Facili da risolvere | ✓ | ? |
| Facili da verificare | ✓ | ✓ |
| Esempi | Domino[1], Euler, ordinamento, ... | Domino[2], Hamilton, Sudoku, Protein folding, Crittografia, ... |

Università degli Studi di Padova

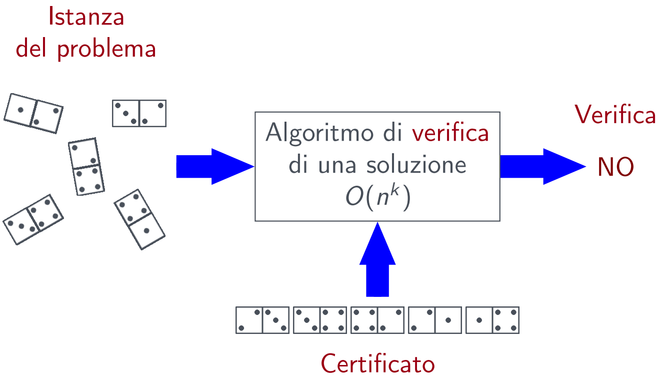


# Domino[2] è in NP



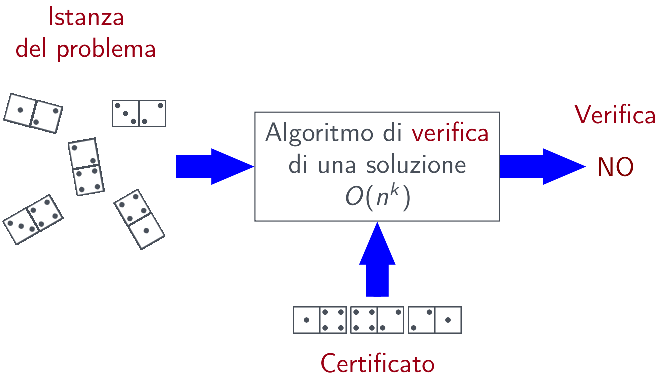
# Domino[2] è in NP





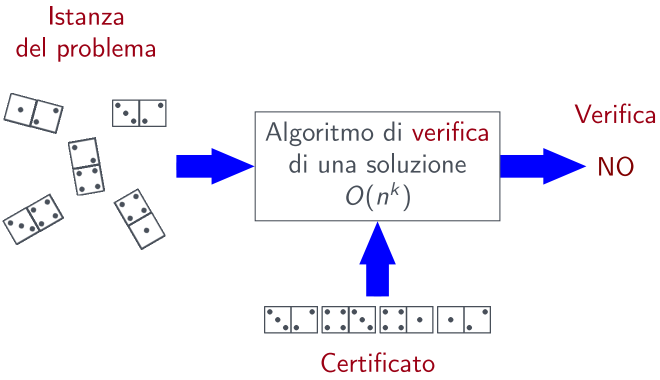
# Domino[2] è in NP





# Domino[2] è in NP





# Pair/Share



- A coppie: risolvete l'esercizio che vi viene consegnato
- A gruppi di quattro: confrontate le soluzioni