

Informatica

e Tecnologie della Comunicazione Digitale

Docente:

Miguel Ceriani (ceriani@di.uniroma1.it)

Lezioni:

Mercoledì/Giovedì/Venerdì 9-11

Ricevimento (su appuntamento):

Mercoledì 14-16 a viale Regina Elena 295, palazzina F, 1° piano

Lezione 15: Linguaggi di Programmazione: Labirinti

Risolvere Labirinti



Claude Shannon, 1952

Risolvere Labirinti

- in informatica spesso studiamo giochi perché rappresentano problemi reali in un contesto controllato e ben definito
- dobbiamo guidare un topo in un labirinto fino a raggiungere una fetta di formaggio
- problemi reali collegati:
 - trovare la strada migliore per raggiungere una destinazione (es., Google Maps)
 - guidare un robot in un ambiente completamente o parzialmente sconosciuto (es., rover spaziali, robot per le emergenze, auto senza conducente)

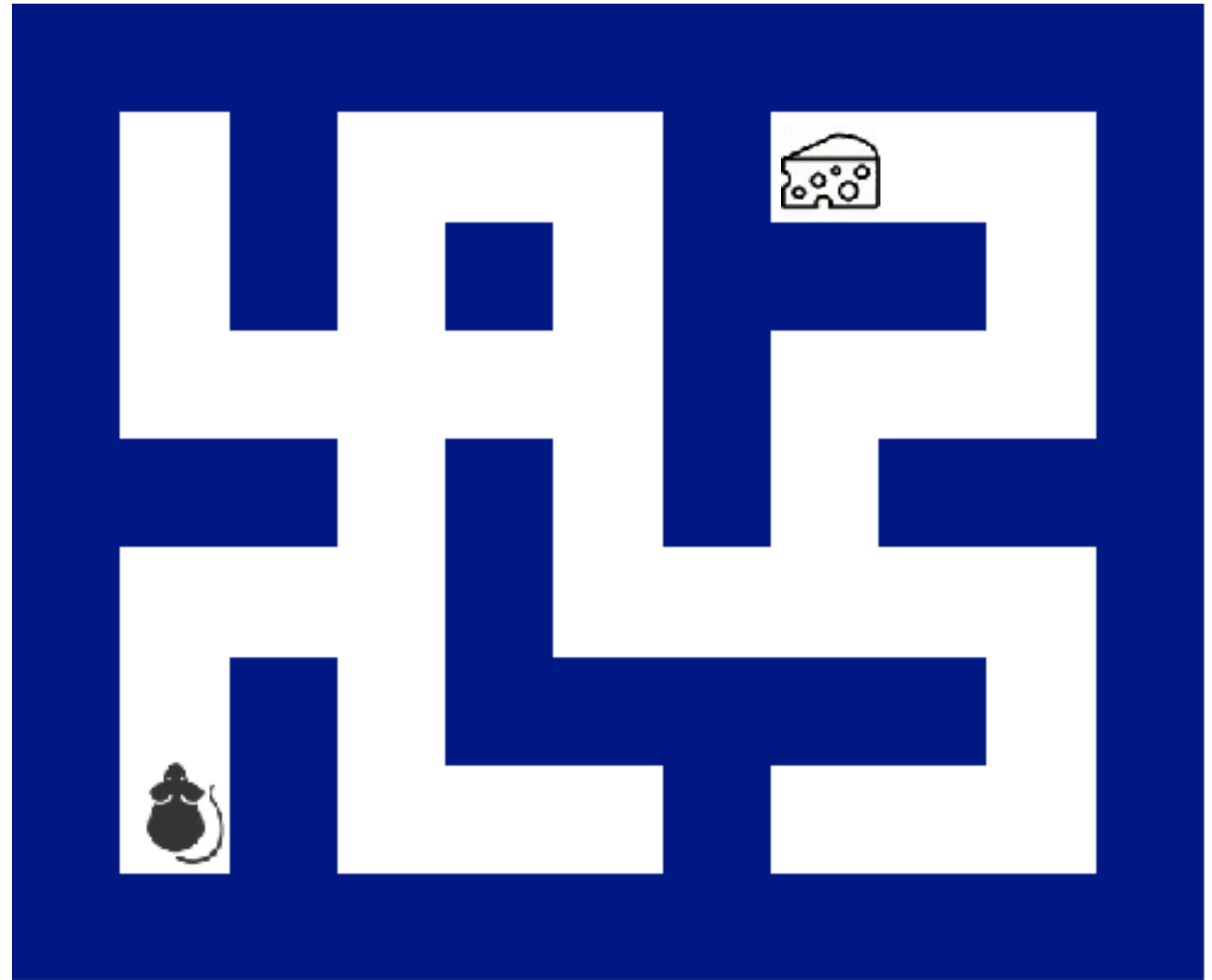
Risolvere Labirinti

Input:

- labirinto
- posizione topo
- posizione formaggio

Output:

- strada (se esiste) che permette al topo di raggiungere il formaggio



Risolvere Labirinti: Due Varianti

1. **Visione Globale**, ho tutte le informazioni:

- configurazione labirinto (per ogni casella tutti i muri che ha attorno)
- posizione/direzione del topo
- posizione del formaggio

2. **Visione Locale**, ho informazioni parziali:

- presenza o meno di muri davanti, a sinistra e a destra del topo
- presenza o meno di formaggio dove si trova il topo

Risolvere Labirinti:

Risoluzione Due Varianti

1. **Visione Globale:**

avendo tutte le informazioni posso calcolare una strada (se esiste), possibilmente ottimale.

2. **Visione Locale:**

devo guidare il topo per esplorare ed acquisire informazioni sul labirinto, fino a trovare il formaggio; il primo tentativo non potrà mai essere ottimale, perché non conosco ancora il labirinto.

Risolvere Labirinti: Variante Locale

- tutte e due le versioni del problema sono interessanti ed hanno applicazioni concrete
- noi da ora in poi analizzeremo solo variante locale

Linguaggio

- stesse istruzioni di base della tartaruga
- diverso insieme di funzioni predefinite

Funzioni Predefinite 1/2:

Azioni

- **avanti()**
fa avanzare il topo di un passo
- **destra()**
fa girare il topo a destra di 90°
- **sinistra()**
fa girare il topo a sinistra di 90°

Valori Booleani

- per adesso abbiamo manipolato valori numerici (numero di passi, gradi, contatori nei cicli for) o stringhe di testo (la lettera che identificava un colore);
- quando per una grandezza possiamo avere solo due possibilità (0 o 1, acceso o spento, vero o falso) lo chiamiamo **valore booleano**;
- ogni valore booleano si può esprimere con una affermazione che è vera o falsa; perciò un valore booleano può assumere valore **vero** o **falso**.

Funzioni Predefinite 2/2:

Informazioni

tutte restituiscono valori booleani:

- **qui_formaggio()**
restituisce vero se il topo si trova dove c'è il formaggio, falso altrimenti
- **strada_avanti()**
restituisce vero se si può andare avanti (non c'è un muro), falso altrimenti
- **strada_destra()**
restituisce vero se si può andare a destra, falso altrimenti
- **strada_sinistra()**
restituisce vero se si può andare a sinistra, falso altrimenti

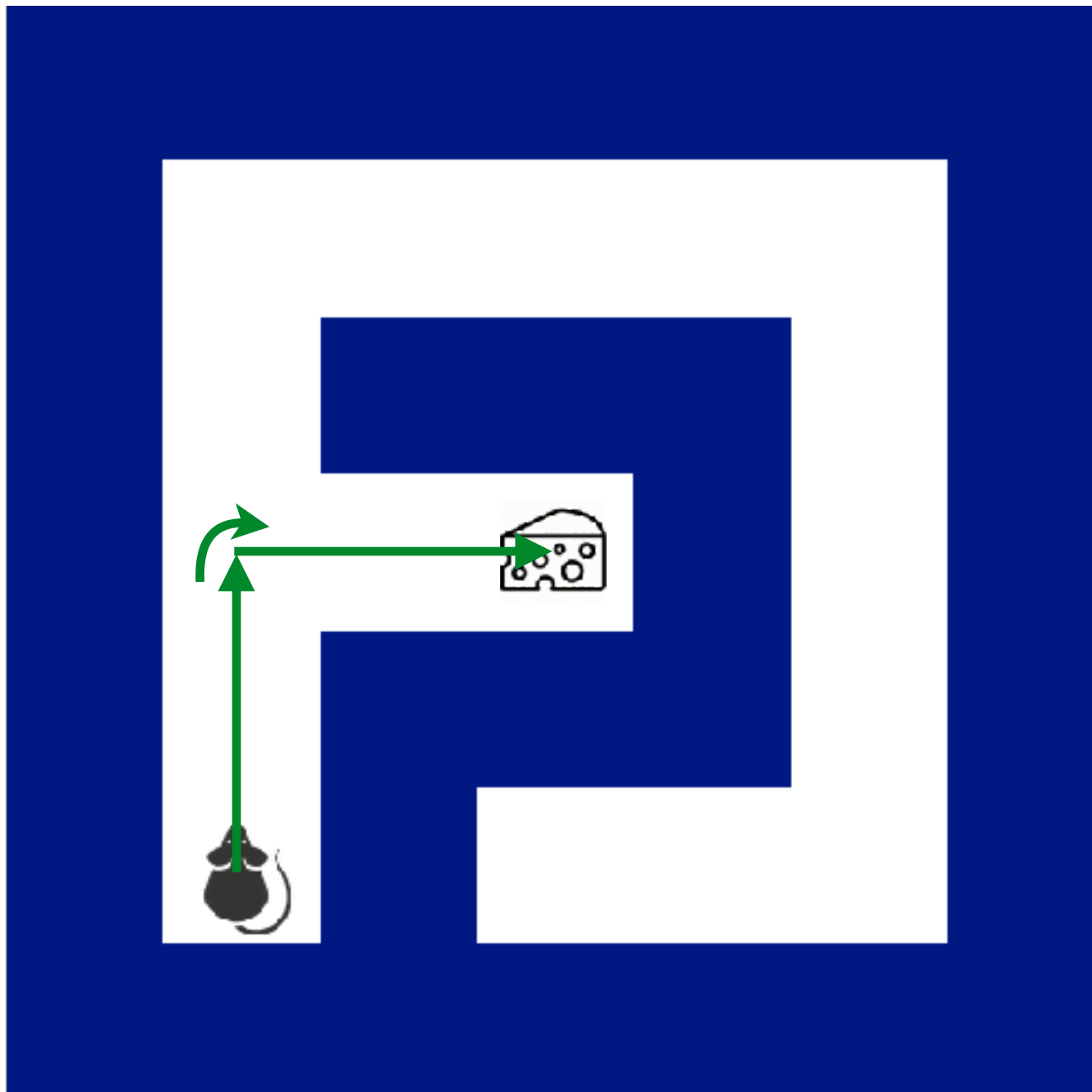
Problemi e Programmi: dal generale al particolare

- **Problema Generale**
scrivere un programma che dato qualunque input (labirinto e posizioni) trova il risultato (porta il topo al formaggio)
- **Problema Particolare**
per una particolare categoria di input (es., tutti i labirinti che hanno solo due curve), scrivere un programma che dato qualunque input in quella categoria trova il risultato
- **Caso Singolo**
per un particolare input, scrivere un programma che lo risolva; li useremo per introdurre il sistema.

Caso Singolo 1



Caso Singolo 1



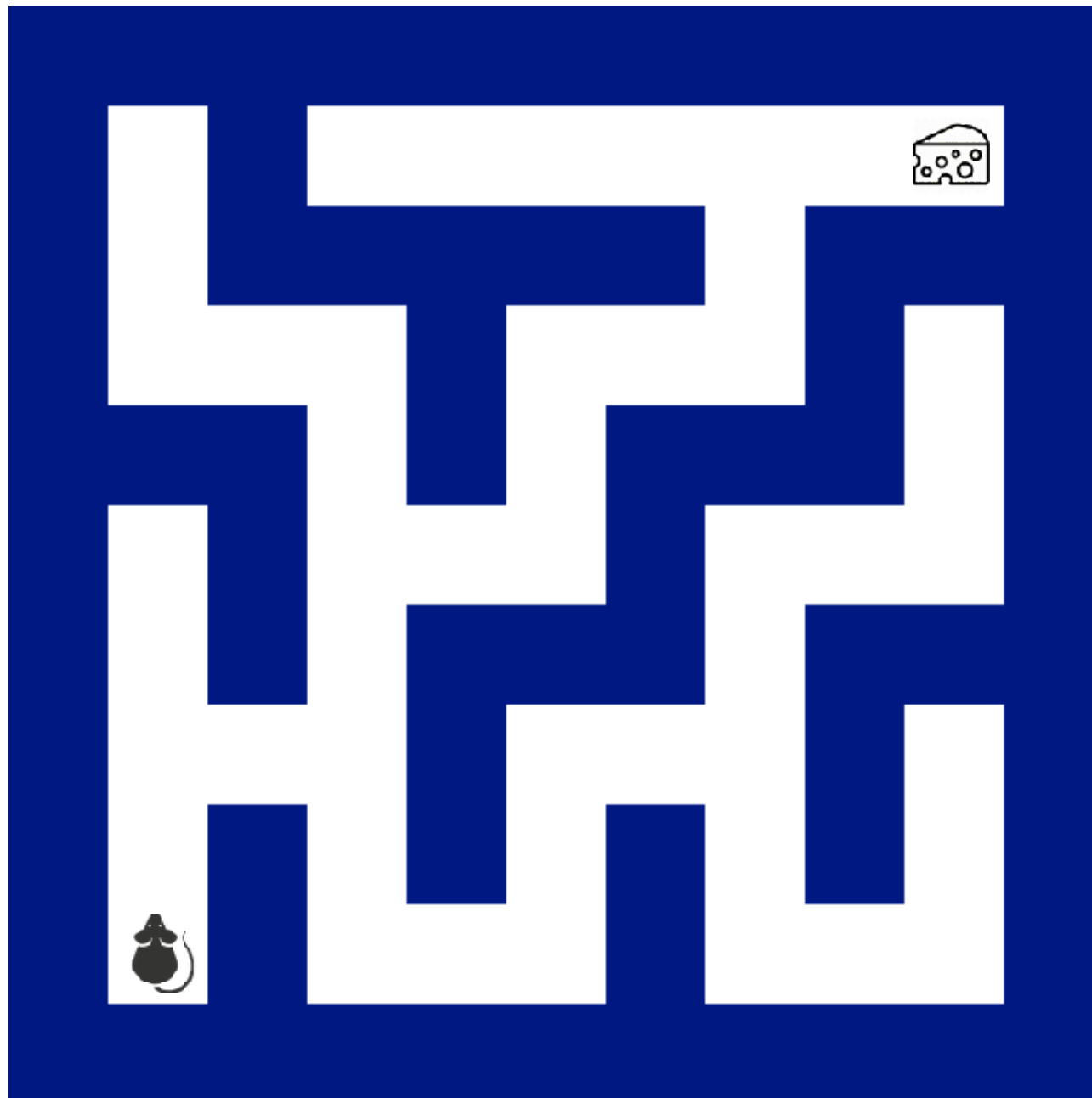
Programma

avanti()

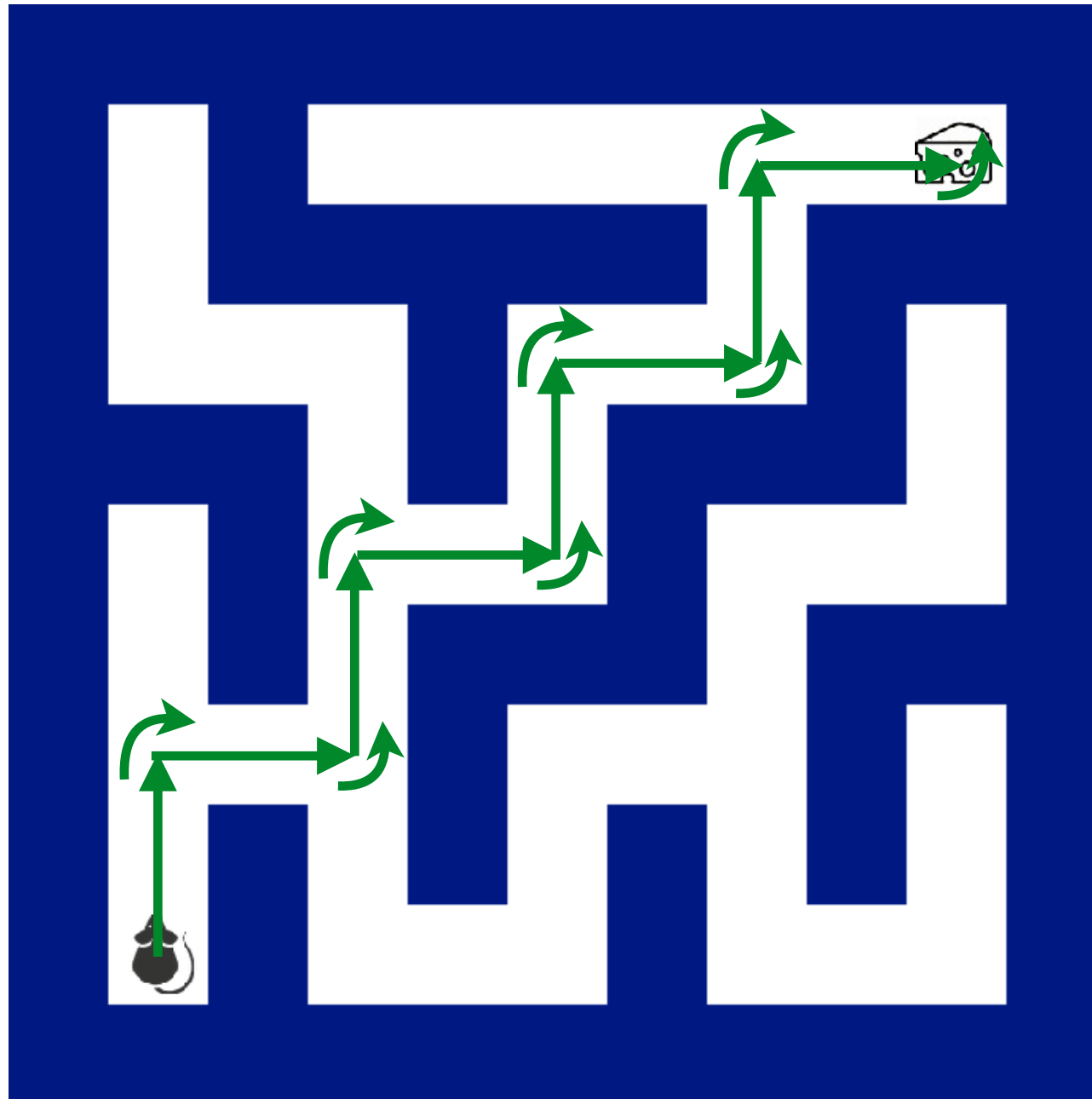
destra()

avanti()

Caso Singolo 2



Caso Singolo 2



Programma

```
RIPETI(4) {
    avanti()
    destra()
    avanti()
    sinistra()
}
```