Giochi di Carte Matemagici

Menezio

28 gennaio 2026

Indice

1 Introduzione

Questo documento riassume *sei* moduli per un laboratorio esteso (circa 8 ore), con enfasi sul funzionamento matematico dei giochi (codifica, permutazioni, basi numeriche, invarianti, permutazioni indotte da mescolamenti). Lo stile è volutamente formale, ma include domande e attività per mantenere l'interazione continua con i partecipanti.

1.1 Obiettivo concettuale

Il messaggio centrale è che molti giochi di carte *auto-funzionanti* sono, in realtà, algoritmi: una sequenza finita di operazioni che, a fronte di input anche parzialmente ignoti, produce sempre un output previsto. L'illusione nasce dal fatto che il pubblico vede un effetto "impossibile", ma non vede la struttura informazionale nascosta.

1.2 Tre idee-chiave

Definizione 1 (Informazione e codifica)**.** *Chiamiamo* informazione *un contenuto che va trasmesso (ad es. "qual è la carta nascosta?"). Una* codifica *è una funzione che associa all'informazione un oggetto osservabile (ad es. l'ordine di alcune carte) in modo che un decodificatore possa recuperare univocamente il messaggio.*

Definizione 2 (Invariante)**.** *Un* invariante *è una proprietà che rimane vera lungo l'esecuzione del procedimento, anche se l'oggetto (mazzo, pila, disposizione) cambia aspetto.*

Definizione 3 (Determinismo vs casualità)**.** *Un procedimento è* deterministico *se, fissati gli input e le regole, l'output è fissato. È* stocastico *se l'output è una variabile casuale (dipende da scelte aleatorie). Molti "trucchi" sembrano stocastici ma sono deterministici.*

Domande guida

1. Cosa significa, concretamente, "nascondere informazione" dentro una permutazione?
2. Qual è la differenza tra "funziona spesso" e "funziona sempre"? Come la dimostriamo?
3. Quando un mescolamento può essere modellato come una permutazione (quindi deterministica)?

Attività in aula

1. **Mini-esercizio (5 minuti).** Quante informazioni posso codificare usando soltanto l'ordine di k oggetti distinti? (Suggerimento: contare le permutazioni.)
2. **Discussione.** In che senso un gioco matemagico è un "protocollo" tra esecutore e assistente (o tra esecutore e pubblico)?

1.3 Un ponte verso i moduli successivi

I sei moduli si possono leggere come variazioni di una stessa idea:

- **Modulo 1:** codifica con permutazioni (Five Card Trick).
- **Modulo 2:** dinamica discreta e punto fisso (21 carte).
- **Modulo 3:** sistemi di numerazione e decomposizione (binario).

- **Modulo 4:** permutazioni e cicli (Solitario dei Re).
- **Modulo 5:** combinatoria dei vincoli (Solitario 1–2–3).
- **Modulo 6:** decomposizione in cicli e strategia (Trova la carta giusta).

2 Modulo 1 — Five Card Trick (Fitch Cheney): codifica con permutazioni

2.1 Descrizione del gioco

Una coppia di maghi propone il seguente gioco al pubblico. Uno dei due maghi esce dalla stanza e una persona del pubblico sceglie 5 carte da un mazzo da 52. Le 5 carte vengono consegnate al mago che è rimasto nella stanza (l'assistente). L'assistente ne mostra **4** al secondo mago, disponendole, in fila, scoperte, in un ordine preciso. Il mago rientra nella stanza e, vedendo solo le 4 carte, identifica la carta nascosta (seme e valore).

2.2 Esempi e casi

2.3 La matematica dietro al trucco

Il trucco è una combinazione elegante di due conteggi:

- (A) **Principio dei cassetti:** tra 5 carte ci sono sempre almeno due carte dello stesso seme.
- (B) **Permutazioni:** l'ordine di 3 carte ha $3! = 6$ possibilità; questo consente di codificare un numero tra 1 e 6.

Partiamo con l'enunciato del **Principio dei cassetti**.

Teorema 1. *Date n oggetti da distribuire in k cassetti, se $n > k$ allora almeno un cassetto contiene più di un oggetto.*

Dimostrazione. Per assurdo, se ogni cassetto contenesse al più un oggetto, il numero totale di oggetti sarebbe al più k, ma ci sono n oggetti con $n > k$, contraddizione. □

Attività in aula

Esercizi sul principio dei cassetti

1. Mostrare che tra 8 oggetti distribuiti in 3 cassetti esiste un cassetto con almeno 3 oggetti.
2. Dimostrare che in un gruppo di 13 persone almeno due condividono lo stesso mese di nascita.
3. Provare che in ogni sequenza di 51 interi esistono due con differenza multipla di 50.
4. (Avanzato) Sia $f : \{1, \dots, 100\} \to \{1, \dots, 49\}$. Dimostrare che esistono quattro indici distinti $a < b < c < d$ tali che $f(a) = f(b)$ e $f(c) = f(d)$, e inoltre $b - a \leq c - a$ e $d - c \leq d - b$, usando solo il principio dei cassetti e argomentazioni combinatorie.

Applichiamo il principio al nostro caso particolare.

Lemma 1 (Due carte dello stesso seme tra 5)**.** *In un mazzo standard con 4 semi, qualunque insieme di 5 carte contiene almeno due carte dello stesso seme.*

Dimostrazione. Per assurdo, se tutte e 5 avessero semi distinti, servirebbero 5 semi diversi. Ma i semi sono 4. Quindi per il principio dei cassetti due carte condividono un seme. ☐

Fissiamo adesso l'ordine ciclico dei 13 valori: $A, 2, 3, \ldots, 10, J, Q, K$ (con A come 1). Date due carte c_1, c_2 dello *stesso seme*, definiamo la distanza "in senso orario"

$$\mathrm{dist}(c_1, c_2) \in \{0, 1, \ldots, 12\}$$

come il numero di passi per andare da c_1 a c_2 lungo il ciclo.

Osservazione 1. *Con la nostra notazione la distanza tra Q e 5 è $\mathrm{dist}(Q, 5) = 6$ (passando per $K, A, 2, 3, 4, 5$).*

Proposizione 1 (Una delle due distanze è abbastanza piccola)**.** *Per due carte distinte dello stesso seme, vale sempre:*

$$\mathrm{dist}(c_1, c_2) \leq 6 \quad \textit{oppure} \quad \mathrm{dist}(c_2, c_1) \leq 6.$$

Dimostrazione. Le due distanze sommano a 13 (perché formano un ciclo completo):

$$\mathrm{dist}(c_1, c_2) + \mathrm{dist}(c_2, c_1) = 13.$$

Se entrambe fossero ≥ 7 avremmo una somma ≥ 14, impossibile. Quindi almeno una delle due distanze è ≤ 6. ☐

Proposizione 2 (Numero di permutazioni di 3 oggetti)**.** *Dati 3 oggetti distinti, esistono esattamente $3! = 6$ modi diversi di ordinarli.*

Dimostrazione. Sono gli anagrammi di una parola con 3 lettere distinte. Pertanto ho $3 \cdot 2 \cdot 1 = 3! = 6$ modi diversi di ordinarli. ☐

2.4 La codifica completa

Step 1: scelta di una coppia stesso seme. Tra le 5 carte, l'assistente individua due carte dello stesso seme: chiamiamole x e y. Queste esistono per il Lemma 2.3.

Step 2: decidere quale nascondere. Scegliamo di mostrare come "carta guida" quella che rende la distanza d verso l'altra *compresa tra 1 e 6*. In concreto:

- se $\mathrm{dist}(x, y) \in \{1, \ldots, 6\}$, allora si **mostra** x e si **nasconde** y;
- altrimenti $\mathrm{dist}(y, x) \in \{1, \ldots, 6\}$, quindi si **mostra** y e si **nasconde** x.

Di nuovo questa scelta è possibile per la Proposizione 2.3 e la distanza $d \in \{1, \ldots, 6\}$ sarà il messaggio numerico da trasmettere.

Step 3: codificare d nell'ordine delle altre tre carte. Delle 5 carte, una è nascosta e una è la guida; restano 3 carte libere. L'assistente decide una corrispondenza biunivoca tra i 6 ordini possibili delle 3 carte e i numeri $\{1, 2, 3, 4, 5, 6\}$. Ad esempio, una possibile corrispondenza potrebbe essere data dalla tabella:

ordine delle 3 carte	d
$(1, 2, 3)$	1
$(1, 3, 2)$	2
$(2, 1, 3)$	3
$(2, 3, 1)$	4
$(3, 1, 2)$	5
$(3, 2, 1)$	6

L'ordine reale sarà scelto secondo la tabella per comunicare d.

Step 4: disposizione finale delle 4 carte viste dal mago. Una scelta comune per codificare e decodificare le informazioni è:

- la **prima** carta è la guida (determina il seme della carta nascosta);
- le **altre tre** sono nell'ordine che codifica d secondo la tabella sopra.

A questo punto l'assistente ha mostrato 4 carte in un ordine preciso. Il mago vede la prima carta, identifica il seme e riconosce, dall'ordine delle tre successive, il numero $d \in \{1, \dots, 6\}$. Quindi ricostruisce la carta nascosta come "stesso seme della guida e valore ottenuto avanzando di d nel ciclo dei 13 valori".

Attività in aula

1. Se il mazzo avesse s semi e r valori per seme, come cambierebbe il progetto?
2. Quale potrebbe essere un modo per codificare in maniera mnemonica l'ordine di 3 carte con i valori da 1 a 6.

2.5 La magia nel reale

Per rendere il trucco più "magico", meno meccanico, e fattibile nel mondo reale proponiamo una codifica e decodifica fatta in questo modo:

- Viene applicato il Lemma dei cassetti e l'assistente sceglie la carta da mostrare come carta guida.
- A seconda del turno che stiamo giocando (cioè il numero di volte n che il trucco è stato fatto), l'assistente posiziona la carta nel posto $n \mod 4$ nella disposizione delle 4 carte mostrate.
- Le altre tre carte vengono disposte in modo da codificare la distanza d con il seguente trucco mnemonico.

    1. Tramite la filastrocca *Come Quando Fuori Piove*, codifichiamo come i semi sono ordinati (Cuori, Quadri, Fiori, Picche). In generale, quindi tutte le carte di cuori sono più piccole di tutte le carte di quadri e così via. All'interno di ogni seme, l'ordine è quello naturale (A,2,3,...,10,J,Q,K).
    2. Abbiamo quindi un ordine totale sulle carte. Chiamiamo G la carta più grande, M la carta di mezzo e P la carta più piccola tra le tre carte rimanenti.
    3. Codifichiamo secondo la tabella seguente:

ordine delle 3 carte	d
(P, M, G)	1
(P, G, M)	2
(M, P, G)	3
(M, G, P)	4
(G, P, M)	5
(G, M, P)	6

L'idea per ricordarsi la tabella è che la prima carta codifica se lo spostamento è piccolo (P, 1 o 2), medio (M, 3 o 4) o grande (G, 5 o 6). Una volta fissata la prima carta, di nuovo la seconda codifica se lo spostamento è il più piccolo o il più grande tra i due sulla base del rapporto tra le altre due carte.

3 Modulo 2 — Gioco delle 21 carte: base 3, funzione discreta e punto fisso

Seppur molto semplice, questo gioco mi ha fatto penare per quanto riguarda la modellizzazione matematica. In realtà, una volta capito il trucco, è molto facile da spiegare, ma non così facile da dimostrare. La formalizzazione discreta di un sistema tipo questo porta tanti piccoli dettagli che vanno curati con attenzione.

3.1 Descrizione del gioco

Si prendano 21 carte qualsiasi di un mazzo. Lo spettatore sceglie mentalmente una carta tra le 21 disponibili. Si distribuiscono le 21 carte in tre colonne (o pile) da 7, distribuendo *a giro*: prima carta nella colonna 1, seconda nella 2, terza nella 3, quarta nella 1, ecc. Lo spettatore indica quale colonna contiene la carta. Si ricompongono le carte mettendo **la colonna indicata in mezzo** alle altre due. Si ripete 3 volte la stessa operazione. Alla fine la carta scelta risulta essere l'**11-esima** del mazzo.

3.2 Formalizzazione della distribuzione

Numeriamo le posizioni del pacchetto dall'alto verso il basso: $1, 2, \dots, 21$. La prima colonna conterrà quindi le carte in posizioni $1, 4, 7, 10, 13, 16, 19$; la seconda quelle in posizioni $2, 5, 8, 11, 14, 17, 20$; la terza quelle in posizioni $3, 6, 9, 12, 15, 18, 21$. Quando si distribuisce in 3 colonne a giro, una carta c che si trova in posizione x finirà nella colonna $C = \text{colonna}(x)$ tale che

$$\text{colonna}(x) \equiv x \pmod 3,$$

e la sua *posizione interna* nella colonna è $\lceil x/3 \rceil$, dove con il simbolo $\lceil \cdot \rceil$ indichiamo la funzione *parte intera superiore* (o *ceil* in inglese), ovvero il più piccolo intero maggiore o uguale al numero dato.

Scriviamo adesso una funzione che presa la posizione di una carta c di una colonna scelta dal giocatore, ci restituisce la nuova posizione di c dopo la ricomposizione del mazzo.

Quando si ricompone mettendo la colonna scelta in mezzo, la nuova posizione della carta dipende dalla sua posizione interna $k \in \{1, \dots, 7\}$:

$$f(k) = 7 + k.$$

Infatti, prima vengono 7 carte (prima colonna non scelta, non importa quale), poi le 7 della colonna scelta, poi le ultime 7.

Poiché $k = \lceil x/3 \rceil$, una modellizzazione pulita della trasformazione è data dalla formula:

$$f : \{1, 2, \dots, 21\} \longrightarrow \{1, 2, \dots, 21\}, \qquad f(x) = 7 + \left\lceil \frac{x}{3} \right\rceil.$$

Per convincerci che questa formula è corretta, può essere utile fare un esempio concreto. Se la carta scelta è in posizione $x = 20$, allora, dopo la scelta della terza colonna e la ricomposizione, la nuova posizione sarà $f(20) = 7 + \left\lceil \frac{20}{3} \right\rceil = 7 + 7 = 14$.

3.3 Il punto fisso e la convergenza

In questa sezione studiamo la dinamica della funzione definita sopra e mostriamo come dopo 3 iterazioni la carta scelta arrivi sempre alla posizione 11.

Proposizione 3 (Punto fisso)**.** *La posizione $x^\star = 11$ è un punto fisso della trasformazione: $f(11) = 11$. Inoltre $f(10) = f(12) = 11$.*

Dimostrazione. Semplice conteggio. $\square$

Per andare avanti nella dimostrazione abbiamo bisogno di una stima sulla funzione "parte intera superiore" e valore assoluto.

Lemma 2 (Stima della funzione "parte intera superiore")**.** *Per ogni* $u, v \in \mathbb{R}$ *vale*

$$\Big|\lceil u\rceil - \lceil v\rceil\Big| \leq \Big\lceil |u-v|\Big\rceil.$$

Dimostrazione. Useremo i seguenti due fatti elementari:

1. (Monotonia) se $a \leq b$ allora $\lceil a\rceil \leq \lceil b\rceil$;
2. (Subadditività) per ogni $x, y \in \mathbb{R}$ vale $\lceil x+y\rceil \leq \lceil x\rceil + \lceil y\rceil$.

Dimostriamo, per completezza, anche la subadditività: poiché $x \leq \lceil x\rceil$ e $y \leq \lceil y\rceil$, si ha

$$x + y \leq \lceil x\rceil + \lceil y\rceil.$$

Applicando la monotonia della funzione $\lceil\cdot\rceil$ otteniamo,

$$\lceil x+y\rceil \leq \Big\lceil\, \lceil x\rceil + \lceil y\rceil \Big\rceil = \lceil x\rceil + \lceil y\rceil,$$

dato che $\lceil x\rceil + \lceil y\rceil$ è un intero.

Ora, ponendo $d := u - v$, applichiamo questi fatti con $u = v + d$. Per monotonia e subadditività,

$$\lceil u\rceil = \lceil v+d\rceil \leq \lceil v\rceil + \lceil d\rceil,$$

cioè

$$\lceil u\rceil - \lceil v\rceil \leq \lceil d\rceil.$$

Scambiando i ruoli di u e v otteniamo anche

$$\lceil v\rceil - \lceil u\rceil \leq \lceil -d\rceil.$$

Quindi

$$-\lceil -d\rceil \leq \lceil u\rceil - \lceil v\rceil \leq \lceil d\rceil.$$

Poiché $\lceil d\rceil \leq \lceil |d|\rceil$ e $\lceil -d\rceil \leq \lceil |d|\rceil$, segue

$$\Big|\lceil u\rceil - \lceil v\rceil\Big| \leq \lceil |d|\rceil = \Big\lceil |u-v|\Big\rceil,$$

come volevamo. □

Il Lemma 2 permette di dimostrare la seguente proprietà di contrazione della funzione f.

Proposizione 4 (Contrazione)**.** *Per ogni* $x \in \{1, \ldots, 21\}$ *vale:*

$$|f(x) - 11| \leq \left\lceil \frac{|x-11|}{3} \right\rceil.$$

In altre parole, la distanza dal numero 11 *di una carta nella colonna scelta, si riduce ad ogni iterazione.*

Dimostrazione. Consideriamo $f(x) = 7 + \left\lceil \frac{x}{3} \right\rceil$, allora $f(x) - 11 = \left\lceil \frac{x}{3} \right\rceil - 4$. Applicando il Lemma 2 con $u = \frac{x}{3}$ e $v = \frac{11}{3}$ otteniamo:

$$|f(x) - 11| = \left|\left\lceil \frac{x}{3} \right\rceil - \left\lceil \frac{11}{3} \right\rceil\right| \leq \left\lceil \left|\frac{x}{3} - \frac{11}{3}\right| \right\rceil = \left\lceil \frac{|x-11|}{3} \right\rceil$$

che è quello che volevamo dimostrare. □

Osservazione 2. *Quello che sta accadendo, è che ogni iterazione* schiaccia *l'informazione sulla posizione centrale dividendo per 3. Alla lunga, scegliendo sempre la colonna con la carta incriminata, la carta converge alla posizione centrale 11.*

A questo punto rimane solamente da mostrare che il procedimento è convergente e che lo fa dopo 3 iterazioni.

Proposizione 5. *Sia x_i la posizione della carta scelta dopo i iterazioni; vale dunque che $f(x_i) = x_{i+1}$. Definiamo $d_i = |x_i - 11|$. Vale allora che, per N sufficientemente grande*

$$d_N \leq 2$$

Dimostrazione. Per la Proposizione 3.3, e usando banalmente che $\lceil x \rceil \leq x + 1$, otteniamo

$$d_{n+1} \leq \left\lceil \frac{d_n}{3} \right\rceil \leq \frac{d_n}{3} + 1$$

e iterando questo otteniamo:

$$d_n \leq \frac{d_0}{3^n} + \sum_{k=0}^{n-1} \frac{1}{3^k} = \frac{d_0}{3^n} + \frac{1 - (1/3)^n}{1 - (1/3)} = \frac{d_0}{3^n} + \frac{3}{2}\left(1 - \frac{1}{3^n}\right).$$

e per n sufficientemente grande si ottiene $d_n \leq 2$. □

Dato che la distanza diventa minore o uguale a 2, applicando nuovamente la disuguaglianza la distanza si riduce a 1 e infine a 0. Quindi dopo un numero finito di iterazioni la carta scelta arriva alla posizione 11.

Osservazione 3. *Il metodo sopra descritto funziona nella generalità e ci dà qualcosa di più forte, cioè la convergenza in un numero finito di passi (stimabile usando la stima fatta, ma non ottimale). Nel caso sopra, la convergenza avviene in 3 iterazioni da conto diretto: inizialmente la distanza massima dalla posizione 11 è 10 (se la carta è in posizione 1 o 21). Dopo la prima iterazione la distanza massima diventa $\lceil 10/3 \rceil = 4$, dopo la seconda diventa $\lceil 4/3 \rceil = 2$, e dopo la terza diventa $\lceil 2/3 \rceil = 1$. La stima dice quindi che dopo tre iterazioni la carta è al massimo a distanza 1 da 11, mentre il conto diretto del gioco garantisce che, in realtà, dopo tre iterazioni la carta è proprio in posizione 11.*

3.4 Generalizzazione: p x q

Il trucco esiste anche come $p \times q$ (con q pile e p carte per pila, spesso con p, q dispari). La logica resta: si "porta in mezzo" la pila indicata e si itera. Si ottiene un punto fisso (la carta converge al centro) e la *base* che governa la convergenza è q (qui $q = 3$). Per approfondimenti si può citare l'interpretazione tramite punti fissi stabili.

Attività in aula

1. Fare una tabella: dopo 1 iterazione la carta può stare solo nelle posizioni 8–14; dopo 2 iterazioni solo in 10–12; dopo 3 iterazioni è forzata a 11.
2. Variante: Cosa succede con 27 carte (3 pile da 9)? Qual è la posizione finale?
3. Variante: progettare un "21 carte con 4 pile" e discutere cosa si rompe.
4. Generalizzazione: per $p \times q$, con p, q entrambi dispari. Siano p le carte da mettere in ognuna delle q pile. Ogni volta che viene scelta una pila, questa viene messa al centro del mazzo e si itera. Qual è la posizione finale della carta scelta? Quante iterazioni servono? La logica rimane la stessa?

4 Modulo 3 — Binary Card Trick: rappresentazione binaria e decodifica per somme

4.1 Descrizione del gioco

Il pubblico sceglie una carta tra le classiche 52 carte. Il mago mostra una serie di insiemi (o "tabelle"): ogni insieme contiene alcune carte. Per ogni insieme il pubblico risponde *sì/no* alla domanda "la tua carta è qui?". Dalla sequenza di risposte, il mago riesce a ricostruire la carta.

4.2 Modello matematico

Per distinguere fino a N possibilità usando risposte sì/no, stiamo costruendo, di fatto, un codice binario. Ogni domanda sì/no fornisce un bit di informazione: "1" se la risposta è sì, "0" se è no. E con m domande sì/no possiamo distinguere fino a 2^m esiti.

Proposizione 6 (Capacità informativa)**.** *Con m domande binarie (risposte sì/no) si possono codificare al più 2^m oggetti distinti.*

Dimostrazione. Ogni sequenza di risposte è una stringa di lunghezza m su alfabeto $\{0, 1\}$. Le stringhe possibili sono 2^m ed è dunque sufficiente associare una informazione diversa a ogni diversa stringa. □

Per costruire il trucco, usiamo la rappresentazione binaria dei numeri. È necessario quindi il seguente teorema.

Teorema 2 (Esistenza ed unicità della rappresentazione in base 2)**.** *Ogni intero x con $0 \leq x < 2^m$ ammette un'unica rappresentazione*

$$x = \sum_{i=0}^{m-1} b_i 2^i, \quad b_i \in \{0, 1\}.$$

Dimostrazione. Dimostriamolo per induzione su m. Per $m = 1$ vale chiaramente: x può essere 0 o 1. Supponiamo ora che valga per m e dimostriamo per $m + 1$. Sia x un intero con $0 \leq x < 2^{m+1}$. Se $x < 2^m$, per ipotesi induttiva esiste una rappresentazione unica

$$x = \sum_{i=0}^{m-1} b_i 2^i.$$

Se invece $x \geq 2^m$, allora possiamo scrivere

$$x = 2^m + y$$

dove $0 \leq y < 2^m$. Per l'ipotesi induttiva, esiste una rappresentazione unica per y:

$$y = \sum_{i=0}^{m-1} b_i 2^i.$$

Quindi

$$x = 2^m + \sum_{i=0}^{m-1} b_i 2^i = \sum_{i=0}^{m} b_i 2^i$$

dove $b_m = 1$ e gli altri b_i sono quelli della rappresentazione di y.

Per quanto riguarda l'unicità, considero due possibili scritture dello stesso numero e la più piccola potenza di 2 che divide una scrittura, ma non l'altra (esiste per ipotesi di due scritture diverse). Allora dividendo per tale potenza, una scrittura diventa dispari, l'altra pari, assurdo. □

Osservazione 4. *Il modo per passare dalla base* 10 *alla base* 2 *fornisce in maniera operativa lo stesso risultato appena dimostrato. L'operazione è quella di dividere ripetutamente per* 2 *fintanto che il quoziente è diverso da zero e prendere i resti. Ad esempio, per rappresentare* 13 *in base* 2*:*

$$13/2 = 6 \quad \textit{resto } 1$$
$$6/2 = 3 \quad \textit{resto } 0$$
$$3/2 = 1 \quad \textit{resto } 1$$
$$1/2 = 0 \quad \textit{resto } 1$$

Leggendo i resti dal basso verso l'alto otteniamo 1101*, che è la rappresentazione binaria di* 13*.*

4.3 Costruzione standard (potenze di 2)

Numeriamo le carte da 1 a N (ad es. $N = 52$). Scegliamo m tale che $2^m \geq N$; per 52 basta $m = 6$ perché $2^6 = 64$.

Definiamo, per $i = 0, \ldots, m-1$, l'insieme

$$S_i = \{\, x \in \{1, \ldots, N\} : \text{il bit } i\text{-esimo di } x \text{ nella rappresentazione binaria è } 1 \,\}.$$

In pratica S_0 contiene i numeri dispari, S_1 quelli con secondo bit 1, ecc.

Mostriamo al pubblico gli insiemi $S_0, \ldots, S_{m-1}$ e chiediamo, per ciascuno, se la carta scelta è presente.

Se il pubblico risponde "sì" all'insieme S_i, allora il bit i della carta è 1. Quindi il mago ricostruisce il numero:

$$x = \sum_{i:\ \text{risposta sì a } S_i} 2^i.$$

Questo è esattamente il teorema di rappresentazione binaria.

Osservazione 5. *Come nel caso del five card trick, usiamo l'ordinamento delle carte dato dal* Come Quando Fuori Piove *per numerarle da* 1 *a* 52*.*

Attività in aula

1. Quante domande servono per indovinare una carta su 100?
2. Che differenza c'è tra numerare le carte e usare direttamente "valore+seme"?
3. Come cambierebbe il trucco in base 3 (domande con 3 risposte)?
4. Costruire i 6 insiemi $S_0, \ldots, S_5$ per i numeri 1 - 52 in un modo che sia facile poi la decodifica.
5. Sapete "rompere" il trucco: cosa accade se un insieme è costruito male (codici non univoci)? Andare verso l'idea di *iniettività* del codice.
6. È possibile creare un trucco simile usando una codifica con la scrittura univoca in base 3? O in base 4? Cosa cambia nel numero di domande necessarie? Basta una risposta "sì/no" o servono più risposte?

5 Modulo 4 — Il solitario dei Re e i cicli delle permutazioni

Entriamo nel mondo dei giochi di carte deterministici con esito probabilistico. Il solitario dei Re è un gioco senza scelte, in cui la vittoria dipende solo dalla disposizione iniziale delle carte. Analizzando il gioco tramite le permutazioni e i loro cicli, si può calcolare la probabilità di vittoria (anche se questo rende il gioco molto meno "magico").

5.1 Descrizione del Gioco

Si utilizza un mazzo da 40 carte.

- Si dispongono 36 carte coperte in 4 file da 9 carte, ogni fila rappresenterà un seme; le 4 carte rimanenti restano in mano.
- Si gira la carta in mano: se è un Re, va in fondo alla fila del suo seme e si pesca la carta che occupava quella posizione; se non è un Re, si va nella posizione indicata (stesso seme e valore), si sostituisce la carta presente e si continua con la nuova carta in mano.
- Il gioco termina in *vittoria* se tutte le carte vengono scoperte; si perde se, una volta posizionati tutti i Re, alcune carte non sono state girate.

5.2 Idea matematica: percorsi su una permutazione

Ogni carta indica dove andare dopo: stiamo iterando una funzione su un insieme finito di 40 posizioni. È naturale cercare di descrivere questo gioco con una *permutazione*, cioè una funzione biiettiva da un insieme di 40 carte in sé stesso. Cerchiamo di definire meglio la struttura combinatoria del gioco e, tramite essa, di calcolare la probabilità di vittoria.

Le 40 carte del mazzo, disposte come richiesto dal problema, possono essere ripensate e descritte come una sequenza di 40 carte costruita come segue:

- La prima carta è la carta in mano all'inizio del gioco;
- La seconda carta è quella indicata dalla carta scoperta al momento della sostituzione della prima carta, la terza carta è quella indicata dalla carta scoperta al momento della sostituzione della seconda carta e così via.
- Se viene trovato un Re, il ciclo si interrompe e si continua a riempire la sequenza di 40 con la carta successiva tra quelle 4 inizialmente in mano.
- Si itera fintanto che non si trovano tutti e 4 i Re.

Con questa costruzione, ogni disposizione iniziale delle carte corrisponde ad una permutazione di 40 elementi, o meglio, per come l'abbiamo identificata noi, a una sequenza di 40 elementi. Infatti, ogni carta indica esattamente una posizione successiva e ogni posizione viene indicata da esattamente una carta.

5.3 Probabilità di vittoria

La domanda che ci poniamo è: *quale caratteristica deve avere la permutazione per garantire la vittoria nel gioco?*

Con questa visualizzazione, la condizione di vittoria del gioco è: *la sequenza di 40 carte individuata dal mazzo deve avere un Re in ultima posizione*.

Quante sono allora le sequenze che verificano la condizione di vittoria? Sono $4 \cdot 39!$ in quanto l'ultima carta può essere uno qualsiasi dei 4 Re e le altre 39 carte possono essere disposte in qualsiasi ordine.

Il numero totale di sequenze possibili è $40!$ in quanto tutte le carte possono essere disposte in qualsiasi ordine.

Quindi la probabilità di vittoria è:

$$\mathbb{P}(\text{vittoria}) = \frac{4 \cdot 39!}{40!} = \frac{4}{40} = \frac{1}{10}.$$

Attività in aula

1. **Versione ridotta.** Provare a riformulare il gioco con carte "pozzo" le carte di cuori. Con quante carte si deve partire? Qual è la probabilità che il gioco venga vinto?
2. **Rappresentazione a grafo.** (avanzato) Supponendo che la partita sia vinta anche se alcune carte rimangono coperte, ma che girandole, risultano nella posizione giusta, come cambia la probabilità calcolata?

6 Modulo 5 — Solitario 1–2–3

6.1 Descrizione del gioco

Sia dato un mazzo classico da 40 carte con 4 semi. Si mescola il mazzo, poi si gira una carta alla volta pronunciando ciclicamente

$$1, 2, 3, 1, 2, 3, \ldots$$

La partita fallisce se, in qualunque istante, la cifra pronunciata coincide con il numero della carta girata; ha successo solo se si arrivano a girare tutte le 40 carte senza coincidenze.

6.2 Primo approccio al problema (computazionale)

Ci piacerebbe capire qual è la probabilità di riuscita del gioco. Un primo approccio potrebbe essere quello di simulare il gioco molte volte e contare quante volte si riesce a completare la partita senza errori. Questo è possibile tramite un piccolo script (Python), riportato qua sotto:

Simulazione Python

```python
import random

def trial():
  deck = []
  for v in range(1, 11):
    deck += [v]*4
  random.shuffle(deck)
  for i, card in enumerate(deck):
    call = (i % 3) + 1
    if card == call:
      return False
  return True

def simulate(n=200000):
  return sum(trial() for _ in range(n)) / n

print(simulate(300000))
```

Il programma simula 300000 partite e calcola la frequenza di successo (ovvero gioca la partita e la conteggia "valida" se non ci sono coincidenze). Eseguendolo, si ottiene una stima della probabilità di riuscita del gioco, che risulta essere circa lo 0.83%.

Le prove sperimentali sono molto divertenti (e possibili al giorno d'oggi), ma in generale non forniscono una comprensione profonda del fenomeno. Tuttavia, permettono di farsi un'idea della difficoltà del gioco e di qualche intuizione. Per una comprensione più profonda, è necessario un approccio teorico.

6.3 Struttura combinatoria dei vincoli

La sequenza pronunciata è periodica: compaiono 14 “uno”, 13 “due” e 13 “tre”. Nel mazzo le carte problematiche sono solo le 12 carte $\{1, 2, 3\}$; le carte 4–10 sono trasparenti ai vincoli (sono in sostanza tutte uguali e non hanno importanza ai fini del conteggio). Traducendo:

- Nessuna carta 1 deve cadere nelle 14 posizioni etichettate con 1;
- Nessuna carta 2 nelle 13 posizioni etichettate con 2;
- Nessuna carta 3 nelle 13 posizioni etichettate con 3.

Stiamo contando, in gergo, "permutazioni di un multinsieme con tre famiglie di divieti sovrapposti". Per farlo proponiamo tre approcci. Il primo è un approccio "standard" con l'inclusione–esclusione; il secondo è un approccio più intuitivo e combinatorico, costruito ad hoc. Il terzo è un metodo computazionale esatto, basato sulla programmazione dinamica e catene di Markov. È interessante vedere metodi diversi per lo stesso problema.

6.4 Il principio in Inclusione – Esclusione

Il principio di inclusione – esclusione nasce dall'osservazione che, contando gli elementi dell'unione di due insiemi A e B, la somma $\mathrm{card}(A) + \mathrm{card}(B)$ considera due volte gli elementi comuni. Per correggere l'errore basta sottrarre l'intersezione:

$$\mathrm{card}(A \cup B) = \mathrm{card}(A) + \mathrm{card}(B) - \mathrm{card}(A \cap B).$$

Lo stesso meccanismo si estende a tre insiemi A, B, C: prima si sommano le cardinalità, poi si sottraggono tutte le intersezioni a due a due, infine si riaggiunge l'intersezione tripla (che era stata eliminata troppe volte):

$$\boxed{\mathrm{card}(A \cup B \cup C) = \mathrm{card}(A) + \mathrm{card}(B) + \mathrm{card}(C) - \mathrm{card}(A \cap B) - \mathrm{card}(A \cap C) - \mathrm{card}(B \cap C) + \mathrm{card}(A \cap B \cap C)}$$

In generale, per insiemi $A_1, \ldots, A_n$, il principio afferma

$$\mathrm{card}\Big(\bigcup_{i=1}^{n} A_i\Big) = \sum_{k=1}^{n} (-1)^{k+1} \left(\sum_{1 \le i_1 < \cdots < i_k \le n} \mathrm{card}(A_{i_1} \cap \cdots \cap A_{i_k}) \right),$$

cioè si alternano somme e sottrazioni delle intersezioni di ordine crescente. Questo garantisce che ogni elemento venga contato esattamente una volta, indipendentemente da quante intersezioni appartenga.

Attività in aula

- Dimostrare il principio di inclusione ed esclusione per induzione su n.
- Applicare il principio di inclusione ed esclusione per contare il numero di numeri coprimi con 15 tra 1 e 100.
- Applicare il principio di inclusione ed esclusione per contare il numero di permutazioni di 30 tra 1 e 100.

Per poter svolgere in maniera furba il conteggio, facciamo una prima semplificazione: consideriamo il mazzo fatto solamente di carte $\{0, 1, 2, 3\}$ con 4 copie di ogni carta 1, 2, 3 e 28 copie della carta 0. Le carte 0 sono "neutre" e non influenzano i vincoli.

Considerando gli 1, i 2 e i 3 indistinguibili, il numero totale di permutazioni contate come fossero anagrammi è

$$N_{\text{tot}} = \frac{40!}{28!\,(4!)^3}.$$

Chiamiamo adesso k_1 il numero di carte 1 che cadono in posizioni con chiamata 1, k_2 il numero di carte 2 in posizioni con chiamata 2 e k_3 il numero di carte 3 in posizioni con chiamata 3.

Consideriamo adesso gli insieme A_{k_1,k_2,k_3} delle permutazioni con almeno k_1 coincidenze per le carte 1, almeno k_2 per le carte 2 e almeno k_3 per le carte 3. Vogliamo contare il numero di permutazioni senza coincidenze. Per farlo, usiamo l'inclusione – esclusione su tutti gli insiemi A_{k_1,k_2,k_3} con $k_i \in \{0, 1, 2, 3, 4\}$ e almeno un $k_i > 0$.

In particolare, prendiamo il totale delle configurazioni N_{tot} e sottraiamo tutte le configurazioni con almeno una coincidenza (attenzione: in quelle con almeno una coincidenza contate in questo modo, sto togliendo più volte quelle che hanno una doppia coincidenza comune), poi aggiungiamo quelle con almeno due coincidenze (che erano state tolte due volte), sottraiamo quelle con almeno tre coincidenze, e infine aggiungiamo quelle con quattro coincidenze (che erano state tolte troppe volte).

Contiamo adesso in maniera ordinata le configurazioni con almeno k_1 coincidenze per le carte 1, k_2 per le carte 2 e k_3 per le carte 3.

- Scegliamo le posizioni delle coincidenze: ci sono $\binom{14}{k_1}$ modi per scegliere le posizioni delle carte 1 tra le 14 disponibili, $\binom{13}{k_2}$ modi per le carte 2 e $\binom{13}{k_3}$ modi per le carte 3.

$$\binom{14}{k_1}\binom{13}{k_2}\binom{13}{k_3}$$

- Rimangono $40 - K$ carte da disporre, dove $K = k_1 + k_2 + k_3$ sono le carte già posizionate. Tra queste, ci sono $4 - k_1$ carte 1, $4 - k_2$ carte 2, $4 - k_3$ carte 3 e 28 carte 0. Tali carte rimanenti possono essere disposte in
$$\frac{(40-K)!}{28!\,(4-k_1)!(4-k_2)!(4-k_3)!}$$
modi distinti anagrammandole.

Applicando il principio di inclusione–esclusione (sottraendo al totale le configurazioni con almeno una coincidenza) otteniamo il numero di permutazioni senza coincidenze.

$$N_{\text{ok}} = \sum_{k_1=0}^{4}\sum_{k_2=0}^{4}\sum_{k_3=0}^{4}(-1)^K\binom{14}{k_1}\binom{13}{k_2}\binom{13}{k_3}\frac{(40-K)!}{28!\,(4-k_1)!(4-k_2)!(4-k_3)!}.$$

Il termine con $K = 0$ coincide con N_{tot}, mentre gli altri sottraggono (o riaggiungono) le configurazioni con almeno una coincidenza.

6.5 Stima numerica e lettura didattica

La probabilità di successo, stimata al computer è

$$p = \frac{N_{\text{ok}}}{N_{\text{tot}}} \approx 0.008307,$$

cioè circa lo 0.83%: serve in media una partita su 120 per riuscire.

Questo è un esempio concreto di quanto l'inclusione–esclusione possa quantificare eventi rari non intuitivi.

Attività in aula

1. Provare una versione ridotta con 12 carte, ciclo 1-2-3 ripetuto 4 volte e vedere se le simulazioni manuali si avvicinano al calcolo teorico.
2. **Inclusione–esclusione a mano.** Provare a espandere esplicitamente la somma con $k_i \in \{0, 1\}$ per capire come nascono i termini positivi/negativi.

6.6 Conto diretto con matrici (Soluzione proposta da David Vencato)

Di seguito riportiamo in maniera testuale la soluzione di *David Vencato*, ex studente del dipartimento di Matematica e attuale PhD ad Oslo.

Le carte possono dividersi in 4 gruppi: gli "uni" (U), i "due" (D), i "tre" (T) e le rimanenti (A). Questa partizione, per come è strutturato il gioco, dà informazioni sufficienti per sapere se il gioco finisce oppure no. Per semplicità, supponiamo che il giocatore termini sempre il mazzo, anche se sa di avere perso.

In particolare, possiamo pensare a una partita come una stringa di 40 lettere composta da 4 "U", 4 "D", 4 "T" e 28 "A" e dunque le combinazioni totali sono date dalla formula per "anagrammi con lettere ripetute":

$$C_T := \frac{40!}{4!\,4!\,4!\,28!}$$

Adesso cerchiamo di capire i casi vincenti. Per fare questo pensiamo a una matrice 3×3:

$$M := \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$$

dove il posto m_{ij} rappresenta il numero di volte che la carta "i" è uscita mentre veniva chiamato il numero "j".

Ad esempio $m_{23} = 1$ vuol dire che esattamente una volta abbiamo estratto un "due" mentre dicevamo "tre".

Affinché M rappresenti una partita possibile sono condizioni necessarie e sufficienti:

- $m_{ij} \in \{0, 1, 2, 3, 4\}$ (non ci sono più di 4 "U" o 4 "D" o 4 "T");
- $\sum_{j=1}^{3} m_{ij} = 4$ (vengono estratti tutti e quattro gli uni, i due e i tre nel corso della partita).

Con questa formulazione è evidente che le matrici vincenti M sono quelle che hanno la diagonale nulla, cioè

$$m_{ii} = 0 \quad \text{per ogni } i = 1, 2, 3.$$

Inoltre, la seconda condizione precedente si riscrive come:

$$\begin{cases} m_{12} + m_{13} = 4 \\ m_{21} + m_{23} = 4 \\ m_{31} + m_{32} = 4 \end{cases}$$

Dunque, tutte e sole le matrici vincenti sono della forma:

$$M = \begin{pmatrix} 0 & m_{12} & 4 - m_{12} \\ m_{21} & 0 & 4 - m_{21} \\ m_{31} & 4 - m_{31} & 0 \end{pmatrix} \quad \text{con } m_{12}, m_{21}, m_{31} \in \{0, 1, 2, 3, 4\}.$$

Dunque, cerchiamo di calcolare quante stringhe sono vincenti data una matrice vincente M.

Con lo stesso ragionamento dell'inizio consideriamo la "sottostringa" data dalle posizioni $1, 4, 7, \dots, 40$, cioè le posizioni in cui viene detto "uno". È una sottostringa di 14 lettere formata da m_{21} "D", m_{31} "T" e $14 - m_{21} - m_{31}$ "A", cioè:

$$\frac{14!}{m_{21}!\, m_{31}!\, (14 - m_{21} - m_{31})!} \quad \text{possibilità.}$$

Allo stesso modo la sottostringa di quando viene detto "due" è formata da 13 lettere con m_{12} "U", $4 - m_{31}$ "T" e

$$13 - m_{12} - (4 - m_{31}) = 9 + m_{31} - m_{12}$$

"A". Quindi:

$$\frac{13!}{m_{12}!\, (4 - m_{31})!\, (9 + m_{31} - m_{12})!} \quad \text{possibilità.}$$

Infine, la sottostringa di quando viene detto "tre" è formata da 13 lettere con $4 - m_{12}$ "U", $4 - m_{21}$ "D" e

$$13 - (4 - m_{12}) - (4 - m_{21}) = 5 + m_{12} + m_{21}$$

"A". Dunque:

$$\frac{13!}{(4 - m_{12})!\, (4 - m_{21})!\, (5 + m_{12} + m_{21})!} \quad \text{possibilità.}$$

Dato che ogni matrice vincente rappresenta sottoinsiemi disgiunti di partite vincenti, il numero totale di stringhe vincenti è:

$$C_V = \sum_{m_{12}, m_{21}, m_{31} = 0}^{4} \frac{14!}{m_{21}!\, m_{31}!\, (14 - m_{21} - m_{31})!} \cdot \frac{13!}{m_{12}!\, (4 - m_{31})!\, (9 + m_{31} - m_{12})!} \cdot \frac{13!}{(4 - m_{12})!\, (4 - m_{21})!\, (5 + m_{12} + m_{21})!}$$

La probabilità cercata è:

$$p = \frac{C_V}{C_T} = \frac{1608107296510}{19358447308200} \approx 0.008307005541819934.$$

6.7 Modellizzazione del solitario 1–2–3 come processo di Markov

Il solitario 1–2–3 può essere modellizzato in modo naturale come un *processo aleatorio a passi discreti* il cui comportamento futuro dipende unicamente dalla configurazione attuale del sistema, e non dalla storia precedente. Questo permette di interpretare il gioco come una *catena di Markov* a stati finiti, rendendo possibile un approccio di programmazione dinamica.

In ogni istante del gioco, ciò che conta per determinare l'evoluzione futura è esclusivamente:

- quali carte restano nel mazzo;
- quale numero verrà chiamato al prossimo passo (1, 2 oppure 3).

L'ordine con cui si è arrivati a tale configurazione non ha alcuna influenza sulle probabilità future. Di conseguenza, il processo soddisfa la *proprietà di Markov*: il futuro dipende solo dallo stato presente.

Formalmente, uno *stato* del gioco è una coppia

$$s = (\mathbf{c}, m),$$

dove:

- $\mathbf{c} = (c_1, c_2, c_3, c_{\text{other}})$ è un vettore di interi non negativi, in cui c_v rappresenta il numero di carte di valore v ancora presenti nel mazzo ;
- $m \in \{0, 1, 2\}$ è un indice che codifica la prossima chiamata, secondo la regola

$$\text{chiamata} = m + 1.$$

Ogni componente c_1, c_2, c_3 assume valori compresi tra 0 e 4. Il valore c_{other} assume valori compresi tra 0 e 28. La somma

$$c_1 + c_2 + c_3 + c_{\text{other}}$$

rappresenta il numero totale di carte rimanenti.

Lo stato iniziale del gioco è quindi dato da

$$\mathbf{c}_0 = (4, 4, 4, 28), \qquad m_0 = 0,$$

che corrisponde alla chiamata iniziale del numero 1.

Definizione 4 (Probabilità di vittoria)**.** *Per ogni stato $s = (\mathbf{c}, m)$ definiamo la funzione*

$$P(s)$$

come la probabilità di completare il mazzo senza mai incorrere in una collisione, *partendo dallo stato s e proseguendo il gioco in modo casuale.*

In particolare:

- *se non restano carte nel mazzo, il gioco è già stato vinto e vale*

$$P(s) = 1;$$

- *se durante un'estrazione viene pescata una carta uguale alla chiamata corrente, la partita termina immediatamente con una sconfitta e quel ramo contribuisce con probabilità nulla.*

Transizioni tra stati

Sia $s = (\mathbf{c}, m)$ uno stato con

$$T = c_1 + c_2 + c_3 + c_{\text{other}} > 0$$

carte rimanenti, e sia $\text{call} = m + 1$ la chiamata corrente.

Alla prossima estrazione, ogni valore v tale che $c_v > 0$ può essere pescato con probabilità

$$\mathbb{P}(\text{pesco } v) = \frac{c_v}{T}.$$

- Se $v = \text{call}$, il gioco termina immediatamente con una sconfitta.
- Se $v \neq \text{call}$, si transita nello stato successivo

$$s' = (\mathbf{c}', m'),$$

dove:

$$\begin{aligned} c'_v &= c_v - 1, \\ c'_u &= c_u \quad \text{per } u \neq v, \\ m' &= (m + 1) \bmod 3. \end{aligned}$$

Combinando le considerazioni precedenti, la probabilità di vittoria soddisfa la seguente equazione di ricorrenza:

$$\mathbb{P}(\mathbf{c}, m) = \sum_{\substack{v=1,2,3,\text{other} \\ v\neq \text{call}}} \frac{c_v}{c_1 + c_2 + c_3 + c_{\text{other}}} \cdot \mathbb{P}(\mathbf{c}^{(v)}, (m+1) \bmod 3),$$

dove $\mathbf{c}^{(v)}$ indica il vettore ottenuto decrementando di uno la componente c_v.

Ogni transizione riduce di una unità il numero totale di carte rimanenti. Di conseguenza, per il grafo degli stati non esistono cicli e ogni sequenza di transizioni termina necessariamente in uno stato terminale.

Questo garantisce che la definizione di $P(s)$ è ben posta e che il calcolo può essere effettuato tramite programmazione dinamica con memorizzazione. Sostanzialmente si aprono ricorsivamente tutte le possibili strade che portano ad avere 0 carte e una volta raggiunta la fine, tutti gli stati vengono ricalcolati partendo dal basso e sommando. Inutile dire che la probabilità di vittoria iniziale calcolata è:

$$\mathbb{P}(\text{vittoria}) = \mathbb{P}(\mathbf{c}_0, m_0) \approx 0.008307005541819934.$$

Concettualmente parlando è importante sottolineare che non si sta calcolando la probabilità di una singola sequenza di carte, ma si associa a ogni configurazione una misura di "quanto essa sia promettente" per la vittoria. Stati identici dal punto di vista delle carte rimanenti e della chiamata futura condividono lo stesso destino probabilistico, indipendentemente dal percorso con cui sono stati raggiunti.

Questo cambio di prospettiva consente di comprimere l'enorme albero delle sequenze possibili in un numero gestibile di stati, rendendo il problema trattabile in modo esatto (ovviamente tramite l'utilizzo di un computer) e ci dà anche la possibilità di sapere la possibilità di vittoria in ogni momento (anche se anche con gli altri metodi, ovviamente, era possibile). Per chi fosse interessato, nel file *app.py* è disponibile la funzione ricorsiva che implementa quanto detto.

Attività in aula

- Come cambia la probabilità di vittoria se invece che chiamare $1, 2, 3$ chiamassi una sequenza diversa? Magari solo $1, 2$ o $1, 2, 3, 4$? Provare a vedere se è possibile trovare un massimo.
- Se si potesse scegliere quando dire i numeri $1, 2, 3$ (sempre 14 volte 1, 13 volte 2 e 13 volte 3) invece che in ordine ciclico, come si potrebbe fare per massimizzare la probabilità di vittoria?

7 Modulo 6 — Trova la carta giusta

7.1 Descrizione del gioco

Due condannati a morte hanno la possibilità di salvarsi. Il Boia spiega un gioco e lascia il tempo ai condannati di concordare una strategia.

- Il Boia fa entrare il primo in una stanza e dispone un mazzo di 40 carte in fila, scoperte, sul tavolo, nell'ordine che preferisce.
- Il condannato può scegliere se effettuare uno scambio tra due carte oppure lasciare tutto come è.

- Dopo la (eventuale) mossa le carte vengono coperte, esce il primo condannato ed entra il secondo.
- Il Boia afferma: "Puoi girare al più 20 carte nell'ordine che preferisci. Se trovi il 2 di Bastoni siete liberi."

È possibile che ci riescano?

7.2 Osservazioni iniziali

- Il primo non sa quale carta verrà richiesta al secondo.
- È più comodo pensare le carte numerate da 1 a 40 invece che con i semi.

7.3 Idea matematica: decomposizione in cicli

Per la soluzione, il primo condannato guarda la stringa di carte:

$$\begin{array}{r|cccc} \text{Num carta} & i_1 & i_2 & \cdots & i_{40} \\ \text{Pos} & 1 & 2 & \cdots & 40 \end{array}$$

Inizia a creare cicli nella sua testa nel seguente modo: prende i_1 e va a vedere cosa trova nella posizione i_1. Troverà i_{i_1}. Allora prende i_{i_1} e va a vedere la carta in posizione i_{i_1} iniziando a scrivere il seguente ciclo

$$(i_1, i_{i_1}, i_{i_{i_1}}, \dots)$$

Il ciclo si chiuderà non appena verrà trovata la carta con $i_j = 1$ poiché dovrei andare a vedere la posizione $i_{i_j} = i_1$ che ho già visto e ripercorrere lo stesso ciclo. Ottengo il ciclo

$$(i_1, i_{i_1}, \dots, 1)$$

Faccio le stesse cose con tutte le carte che NON ho visto nel primo ciclo e spezzetto le 40 carte in cicli.

Osservazione 6. *Se con questo ragionamento parto dalla posizione s, cioè parto da i_s, allora dentro al suo ciclo ci sarà la carta s.*

7.4 Strategia

Supponiamo adesso che i cicli che costruisco sono tutti di lunghezza ≤ 20. Allora il primo condannato non fa niente. Il secondo, partendo dalla posizione della carta indicata dal Boia e percorrendo il suo ciclo, troverà la carta del Boia entro la fine del ciclo, cioè entro le ≤ 20 carte girate.

Se invece c'è un ciclo lungo > 20, allora questo ciclo è l'unico di lunghezza > 20 (poiché $40 - (\text{cicli}) \leq 19$). Considero quindi il ciclo in questione:

$$(j_1, \dots, j_t) \quad \text{con } 20 < t \leq 40$$

Considero la carta

$$k = \begin{cases} j_{t/2} & \text{se } t \text{ pari} \\ j_{(t+1)/2} & \text{se } t \text{ dispari} \end{cases}$$

Inverto la carta k con la carta j_t. A questo punto, provando a ripercorrere il ciclo cosa accade? Chiamo $k-1$ la carta del ciclo precedente a k:

$$(j_1, j_2, \dots, k-1)$$

Arrivati qui devo andare a vedere che carta c'è in posizione $k-1$. Per lo scambio fatto c'è j_t. Ma la carta indicata da j_t è proprio j_1 e il ciclo si chiude. Ho dunque un ciclo

$$(j_1, j_2, \dots, k-1, j_t)$$

che per costruzione è lungo

$$\begin{cases} t/2 & \text{se } t \text{ pari} \\ (t+1)/2 & \text{se } t \text{ dispari} \end{cases}$$

Analogamente, chiamando $k+1$ la carta dopo k la seconda parte del ciclo sarà

$$(k+1, \dots, j_{t-1}, k)$$

che è un ciclo lungo

$$\begin{cases} t/2 & \text{se } t \text{ pari} \\ (t+1)/2 & \text{se } t \text{ dispari} \end{cases}$$

Tramite lo scambio sono riuscito a spezzare l'unico ciclo che mi dava problemi e l'ho spezzato in 2 cicli di lunghezza ≤ 20. Dunque la strategia del secondo funzionerà e siamo salvi.

7.5 Estensioni

- Si può fare con N carte generiche.
- Si può aumentare il numero di scambi.

Dovrebbe valere che se ho a disposizione k scambi allora è sufficiente girare $\lceil N/k \rceil$ carte per chiudere (è coerente col fatto che con $k = N$ scambi ne basta 1).

Riferimenti essenziali (facoltativi)

Per chi vuole una bibliografia minima, si possono citare:

- Analisi e materiali didattici sul Five Card Trick e codifica con permutazioni.
- Analisi del 21-card trick come funzione discreta con punto fisso stabile.
- Definizioni e descrizione del Gilbreath shuffle/principle come famiglia di permutazioni con vincoli.

(Se vuoi, posso aggiungere una sezione `thebibliography` completa con riferimenti in stile accademico.)

Chiusura (integrazione trasversale)

Mappa concettuale dei moduli

Modulo	Oggetto matematico	Parole chiave
1	permutazioni finite	$3! = 6$, distanza modulo 13
2	dinamica discreta	funzione, punto fisso, base 3
3	codifica binaria	2^m, rappresentazione, decodifica
4	permutazioni e cicli	probabilità, strutture di cicli
5	vincoli combinatori	inclusione–esclusione, catene di Markov
6	decomposizione in cicli	strategia, scambio

Proposta di mini-progetto finale (30–45 min)

Attività in aula

Dividere la classe in piccoli gruppi e chiedere di progettare un "nuovo" effetto con vincoli:

1. deve esserci una **codifica** (ordine o scelta di pile) e una **decodifica** verificabile;
2. deve esserci una **dimostrazione** (anche informale ma corretta) del perché funziona sempre;
3. il gruppo deve presentare una "spiegazione pubblica" (senza rivelare dettagli superflui) e una "spiegazione matematica" completa.