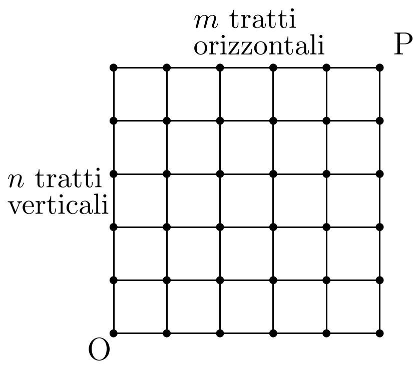
# Indice

# 1 Scheda n. 1d: il problema "Manhattan"

In questa scheda viene presentato il percorso didattico introduttivo al problema "Manhattan". A partire dal testo del problema, viene derivata e discussa una possibile rappresentazione simbolica del problema stesso. Analizzando alcuni semplici modelli, i discenti verranno guidati dal docente ad affrontare e risolvere alcune questioni preeliminari, connesse al problema, ma anche a riflettere sui temi generali come la *rappresentazione della conoscenza* e *l'algoritmo*, come soluzione di un problema.

**Testo del problema**
Considera una mappa stradale come quella rappresentata nella figura sottostante, che ricorda le strade del quartiere di New York "Manhattan".




I punti rappresentano gli incroci, mentre i tratti orizzontali e verticali, tutti della stessa lunghezza $L$, rappresentano le strade che uniscono questi incroci. Osserva che gli incroci $O$ e $P$ riportati nella figura, rappresentano i due vertici opposti di un rettangolo di base $mL$ ed altezza $nL$. Vogliamo conoscere:

1. qual'è, se esiste, la minima lunghezza di un percorso che, rimanendo all'interno del rettangolo, parta dal punto $P$ e giunga al punto $O$;

2. se tale minima lunghezza esiste, quanti sono i possibili percorsi distinti di minima lunghezza.

**Spunti di riflessione per i discenti** [discussione]

- Il problema proposto, è connesso alla scelta del percorso più conveniente, nel senso di tempo di percorrenza, per raggiungere un dato incrocio in una rete stradale analoga a quella del quartiere di New York "Manhattan". Perché?

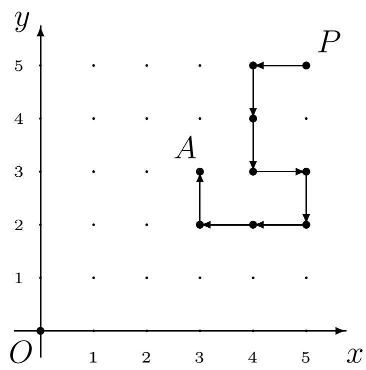
- La lunghezza dei percorsi da $P$ ad $O$ è sempre superiore alla distanza a volo d'uccello? Se no, in quali casi?

- Esiste realmente tra i percorsi un cammino minimo?

**Rappresentazione del problema nel piano cartesiano** [ lezione frontale]

Per affrontare il problema, occorre rappresentarlo in maniera formale in un particolare dominio, o modello.

Per fare ciò, si può ricorrere ad un piano cartesiano $Oxy$ con prefissata unità di riferimento $u$ pari a $L$ metri. In questo piano, la meta $O$ dei percorsi viene associata all'origine $O$ del sistema di riferiemento, l'incrocio di partenza con un punto $P$ di coordinate $(m, n)$ nel primo quadrante, mentre tutti gli altri incroci vengono rappresentati dai punti con coordinate intere all'interno del rettangolo di vertici $P$ ed $O$. Un esempio di tale rappresentazione, per $m = n = 5$, è riportato nella figura sottostante.




Un percorso nella rete stradale viene rappresento nel piano $Oxy$, con una squenza di segmenti contigui, ciascuno lungo 1, che congiungono il punto di partenza $P$ con il punto di arrivo $O$. D'ora innanzi, ci riferiremo a tale spezzata con il termine *cammino*, per distinguerla dal percorso che essa rappresenta nella rete stradale. Nella figura sopra, ad esempio, è disegnato un cammino che rappresenta un percorso di lunghezza $8L$ (essendo l'unità di riferimento $u$ lunga a $L$) che parte dall'incrocio $P$ e giunge all'incrocio $A$.

**Spunti di riflessione per i discenti** [conversazione clinica, discussione]

- La rappresentazione che abbiamo scelto è adeguata a descrivere il nostro problema?
- Come viene tradotto il problema "Manhattan" nel modello prescelto?

**Suggerimento per il docente** Può essere interessante, a questo punto, spendere qualche parola sulla *rappresentazione della conoscenza* mediante i modelli simbolico-formali, tema caro alla Logica, all'Informatica, ed anche all'Intelligenza Artificiale. Ad esempio, si potrebbe presentare uno schema generale che metta in evidenza l'associazione tra i dati iniziali (o finali) del nostro problema, e la realtiva rappresenzazione nel modello prescelto, rimarcando che le operazioni fatte sul modello corrispondono univocamente a ragionamenti o azioni realtivi alla rete stradale considerata.

**Esercizio da svolgere in classe** [lavoro di gruppo, discussione]
Si individuino tutti i percorsi minimi che congiungono il punto $P(2,3)$ con $O(0,0)$. Successivamente, si rifletta su questi punti

- Come poter essere sicuri di aver individuato dei cammini minimi?
- Come poter essere sicuri di aver individuato tutti e soli i cammini minimi?
- Si è fatta una ricerca nell'iniseme di tutti i possibili cammini, o si è trascurato di proposito i cammini eccessivamente zigzaganti, perché intuitivamente troppo lunghi?

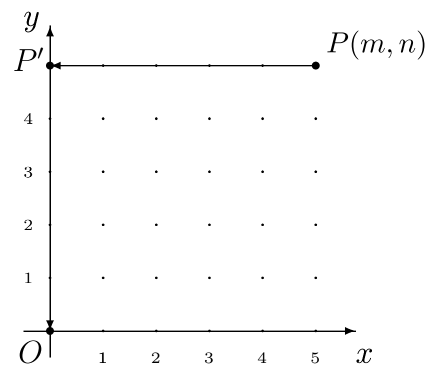
Nell'esempio proposto, l'esistenza di un cammino minimo è stata provata operativamente. In generale, dato un qualunque incrocio della rete stradale, ovvero un arbitrario punto $P(m,n)$ nel primo quadrante del piano cartesiano,

esiste un percorso di lunghezza minima che inizia in $P$
e termina in $O$?

Come vedremo tra breve, la risposta formale a questa domanda, risiede negli assiomi dei numeri naturali.

Innanzitutto, si osservi che la lunghezza di un dato percorso risulta $L$ volte la lunghezza del cammino che lo rappresenta nel nostro piano $Oxy$. Dunque sarà sufficiente chiederci se esiste un qualche cammino, fatto da soli tratti unitari orizzontali e verticali, che sia di lunghezza minima e congiunga i punti $P$ ed $O$.

Ora, cammini che congiungano $P$ con $O$ ne esitono senz'altro; ad esempio quello costituito dai segmenti contigui $PP'$ e $P'O$ rappresentati nella sottostante figura.




Quindi, se consideriamo l'insieme delle lunghezze di tutte i cammini che congiungono $P$ ed $O$, esso risulta essere un sottoinsieme non vuoto dell'insieme $\mathbb{N}$ dei numeri naturali. Dunque, per l'Assioma del Buon Orinamento di $\mathbb{N}$, esisterà un minimo valore di lunghezza, che chiamiamo $\mathrm{d}_{PO}$[1], e quindi una qualche spezzata di lunghezza minima $\mathrm{d}_{PO}$, che rappresenta, nella rete stradale, un percorso di lunghezza minima $L\mathrm{d}_{PO}$.

**Spunti di riflessione per i discenti** [discussione]

[1]In particolare, $\mathrm{d}_{PO}$ si chiama distanza di Manhattan tra i punti P ed O

- Ad ogni punto $P(m, n)$ del primo quadrante siamo quindi in grado di associare il numero $\mathrm{d}_{PO}$, che rappresenta la lunghezza minima dei cammini che uniscono $P$ ed $O$. Che dominio e codominio ha questa una funzione? E' iniettiva, suriettiva?

L'esistenza di cammini minimi per ogni punto $P$ del primo quadrante, permette di definire il problema "Manhattan" in modo formale. Risulta infatti ben definita la funzione

$$\mathbf{M} : \mathbb{N} \times \mathbb{N} \to \mathbb{N},$$

che a un qualunque incrocio della rete, ovvero a un qualunque punto $P(m, n)$ che lo rappresenta, associa il numero intero

$$\mathbf{M}(m, n) = \text{numero dei cammini da } P(m, n) \text{ ad } O \text{ di lunghezza minima.}$$

Risolvere il problema "Manhattan", equivale quindi a determinare, per un arbitrario punto $P(m, n)$, il corrispondente valore $\mathbf{M}(m, n)$. Dunque, la soluzione del problema sarà fornire esplicitamente la legge $\mathbf{M}$, ovvero la descrizione di una sequenza di operazioni computazionali che trasformano un qualunque dato iniziale $(m, n)$ nel dato finale $\mathbf{M}(m, n)$, vale a dire, un *algoritmo*.

**Esercizi da svogersi a casa**

- Riflettere sulla questione dell'iniettività e suriettività della funzione $P \to \mathrm{d}_{PO}$.
- Analizzando dei semplici casi, ad esempio $P(1, 2)$, $P(2, 2)$ e $P(2, 3)$,
  - cosa si può dire sull'unicità dei cammini minimi?
  - provare a dare una qualche descrizione (caratterizzazione) dei cammini minimi;
  - secondo la vostra intuizione[2], dato il punto di partenza $P(m, n)$, quanto vale $\mathrm{d}_{PO}$?

[2]Si considerino i valori $\mathrm{d}_{PO}$ per i punti P più vicini ad $O$.

## 2 Scheda n. 2d: caratterizzazione dei percorsi minimi

In questa scheda viene presentata una caratterizzazione dei cammini minimi, che si basa su semplici ed intuitivi argomenti. Essa fornisce uno strumento importante per risolvere il problema "Manhattan", ma costituisce anche un punto di contatto con argomenti quali l'*algebra dei vettori* e il *gruppo delle traslazioni del piano*.

**Discussione esercizi assegnati per casa nella lezione precedente** [discussione]

- Mostrare con semplici esempi che $P \to \mathrm{d}_{PO}$ non è né iniettività né suriettività, e sembra associare a $P$ la somma delle sue coordinate.
- Rilevare che in generale vi è più di un cammino minimo, ad esclusione dei punti sugli assi coordinati, che ne ammettono solo uno.

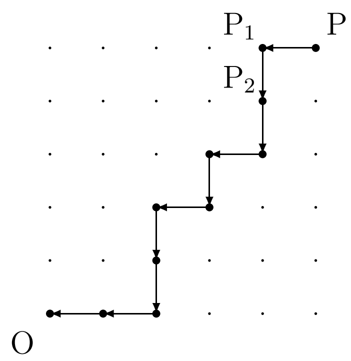
**Caratterizzazione cammini minimi** [ discussione, lezione frontale]
Per cercare uan risposta al problema in esame, è necessario a questo punto trovare una caratterizzazione dei cammini minimi, che colga alcune intuizioni che saranno senz'altro balenate ai discenti risolvendo alcuni casi. La metodologia didattica, da adottare in questa fase, potrebbe alternare momenti di discussione argomentativa con i discenti, in cui vengano raccolte idee e intuizioni, a momenti in cui il docente sintetizza formalmente delle congetture, e ne da una successiva dimostrazione o confutazione.

La rappresentazione di un percorso nella nostra rete stradale può essere fatta utilizzando, invece dei cammini, una sequenza di istruzioni di movimento di una lunghezza $L$ lungo le quattro direzioni cardinali (Nord, Sud, Ovest, Est) a partire dal punto di partenza $P$.

Ad esempio, la sequenza

$$\Sigma = (O,\ S,\ S,\ O,\ S,\ O,\ S,\ S,\ O,\ O)$$

rappresenta il cammini mostrato nella figura sottostante.



Si noti, che il numero di istruzioni di movimento è pari alla lunghezza del cammino, e quindi direttamente proporzionale alla lunghezza del percorso che esso rappresenta. Nell'esempio in figura, il cammino è lungo 10, dunque il percorso che esso rappresenta risulta lungo $10L$.

Vogliamo mostrare ora alcune caratteristiche dei cammini di lunghezza finita che partono da un generico incrocio $P(m,n)$ della rete. Come abbiamo visto poco sopra, un arbitrario cammino è defīnto dalla sequenza $\Sigma = (I_1, \ldots, I_k)$ delle istruzioni di movimento. Ora definiamo i seguenti numeri

$$\begin{aligned}
\Sigma_N &= \text{numero di istruzioni di movimento verso Nord} \\
\Sigma_S &= \text{numero di istruzioni di movimento verso Sud} \\
\Sigma_O &= \text{numero di istruzioni di movimento verso Ovest} \\
\Sigma_E &= \text{numero di istruzioni di movimento verso Est}
\end{aligned}$$

Se il punto di partenza $P$ ha coordinate $(m,n)$, allora è facile provare che il punto di arrivo $A$ è di coordinate

$$A \equiv (m + \Sigma_E - \Sigma_O,\ n + \Sigma_N - \Sigma_S) \tag{1}$$

e che la lunghezza $L_\Sigma$ del percorso è

$$L_\Sigma = L(\Sigma_N + \Sigma_S + \Sigma_E + \Sigma_O) \tag{2}$$

La proprietà (2) è autoevidente, mentre la dimostrazione della relazione (1) non è difficile.

Si osservi che questa proprietà può anche essere ottenuta in termini di associatività e commutatività della somma vettoriale, una volta che le istruzioni di movimento vengano interpretate con vettori spostamento lungo le quattro direzioni possibili.

Analogamente, la proprietà può ricondursi alle proprietà del gruppo delle traslazioni del piano, una volta che ad ogni istruzione di movimento venga associata una opportuna traslazione nella direzione del movimento.

**Suggerimento per il docente** Questa dimostrazione può essere anche lasciata come esercizio per casa, magari dando ai discenti qualche suggerimento, come quelli di seguito.

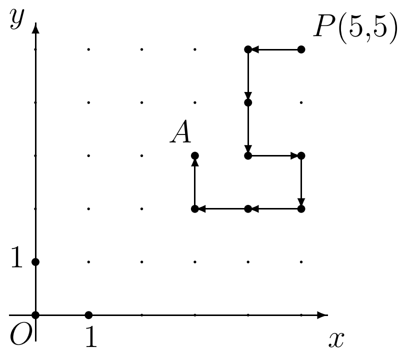
Per dimostrare la proprietà, innanzitutto si osserva che gli spostamenti nella direzione Nord/Sud, che sono tutti e soli quelli responsabili dell'ordinata $n'$ del punto di arrivo $A(m', n')$, e che quelli nella direzione Ovest/Est sono tutti e soli quelli resposabili dell'ascissa $m'$ di $A(m', n')$. Poi si conclude costatando che una coppia di passi l'uno in direzione Nord, l'altro in Sud, danno un contributo nullo in termini di spostamento nella direzione Nord/Sud, per cui la risultante del movimento in quella direzione sarà data dalla differenza $\Sigma_N - \Sigma_S$.

Nell'esempio riportato nella figura sottostante, il punto $P$ ha coordiante (5,5) e il cammino è caratterizzato dalla seguente sequenza di istruzioni:

$$\Sigma = (O,\ S,\ S,\ E,\ S,\ O,\ O,\ N),$$

per cui $\Sigma_N = 1$, $\Sigma_S = 3$, $\Sigma_O = 3$, $\Sigma_E = 1$, e il punto di arrivo $A$ ha coordinate

$$A \equiv (5 + 1 - 3, 5 + 1 - 4) = (3, 3).$$




Le sequenze $\Sigma$ di istruzioni di movimento che realizzano percorsi da $P$ a $O$, sono perciò caratterizzate dalle due relazioni:

$$m + \Sigma_E - \Sigma_O = 0, \quad n + \Sigma_N - \Sigma_S = 0. \tag{3}$$

Tramite queste relazioni si deduce che la lunghezza $l_\Sigma$ di tali cammini non può essere inferiore a $n + m$, infatti

$$l_\Sigma = \Sigma_N + \Sigma_S + \Sigma_E + \Sigma_O = m + n + 2(\Sigma_N + \Sigma_E) \geq m + n \tag{4}$$

A questo punto è facile provare le seguenti due caratterizzazioni dei cammini minimi da $P$ ad $O$:

1. i cammini minimi da $P$ ad $O$ hanno lunghezza $n + m$;

2. i cammini minimi da $P$ ad $O$ sono tutti e soli quelli le cui sequenze di movimento $\Sigma$ che hanno $m$ istruzioni di movimento in direzione Ovest e $n$ istruzioni di movimento in direzione Sud.

Le due affermazioni si dimostrano assieme come segue. Dalla disequazione (4) si sa che la minima lunghezza dei cammini da $P$ ad $O$ non può essere inferiore a $n + m$. D'altra parte, la caratterizzazione (3) e l'equazione (2) ci dicono che i cammini con

$$\Sigma_O = m, \ \Sigma_S = n, \ \Sigma_E = \Sigma_N = 0,$$

sono tutti e soli quei cammini di lunghezza $n + m$ che congiungono $P$ con $O$.
**Esercizi da svogere a casa**

- La caratterizzazione che abbiamo scoperto, come può essere utilizzata per verificare se abbiamo trovato effettivamente tutti i cammini minimi dei casi smplici con $P(1, 2)$, $P(2, 2)$ e $P(2, 3)$, discussi all'inizio della lezione?

- Completare la dimostrazione lasciata come esercizio.
- Creare una tabella con i valori $\mathbf{M}(m, n)$ ottenuti per i punti di una girglia 4x4, cercando di scoprire eventuali regolarità nelle sequenze di numeri ricavate.

# 3 Scheda n. 3d: la soluzione ricorsiva del problema "Manhattan"

In questa scheda viene presentato un possibile percorso didattico attraverso cui guidare i discenti alla scoperta dell'algoritmo ricorsivo che risolve il problema "Manhattan". I concetto informatico più rilevante che viene introdotto sono quelli di *algoritmo ricorsivo* e *funzione ricorsiva.*

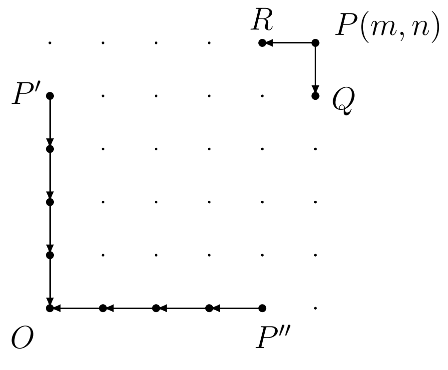
**Discussione esercizi assegnati per casa** [ discussione]

- Mostrare che la caratterizzazione trovata è uno strumento potente per identificare i cammini minimi. Completare assieme ai discenti la dimostrazione lasciata in sospeso;
- è stata riconosciuta la struttura del triangolo di Pascal?

**Studio di una possibile soluzione** [discussione, lezione frontale]
Grazie alla caratterizzazione di cammino minimo, è immediato dimostrare che tutti i punti che giacciono sugli assi coordinati, ammettono un'unico cammino minimo che li congiunge ad $O$. Infatti le uniche direzioni di movimento consentite ai cammini minimi sono Ovest e Sud. Ad esempio, nella figura sottostante $P'$ è origine di un unico cammino minimo che giunge ad $O$, giacente sull'asse $y$, mantre il cammino minimo con origine in $P''$ giace sull'asse $x$.

Diciamo **casi base** i punti di partenza che giacciono sugli assi cartesiani, cioè quei casi in cui il calcolo della funzione $\mathbf{M}$ è immediato; poco sopra, abbiamo visto che per tutti i casi base, la funzione $\mathbf{M}$ vale 1. Riassumendo si ha che

$$\forall m, n \in \mathbb{N} \ : \ \mathbf{M}(m,0) = \mathbf{M}(0,n) = 1 \tag{5}$$



Consideriamo ora un punto del tipo $P(m, n)$ che non giaccia sugli assi cartesiani, cioè con $m, n > 0$. In virtù della caratterizzazione dimostrata nella Scheda 2d, sappiamo che tutti i cammini minimi che partono da $P$ e giungono ad $O$, devono essere fatti di $n$ tratti in direzione Sud e $m$ tratti in direzione Ovest. Quindi, un percorso minimo deve necessariamente passare per il punto $R(m-1, n)$ o per il punto $Q(n, m-1)$ (ma non per tutti e due!).
E' facile provare per assurdo, che un percorso minimo da P a O, passante per $R$ (oppure per $Q$) individua univocamente un percorso minimo da R (oppure da $Q$) ad $O$, e viceversa.

**Suggerimento per il docente** Questa dimostrazione può essere anche lasciata come esercizio per casa.

**Esercizio in classe** [discussione]
Prima di affrontare il caso generale, derivando la regola ricorsiva, è conveniente proporre ai discenti un caso semplice, ad esempio con $P(2, 3)$ ($R(1, 3)$, $Q(2, 2)$), e far calcolare i cammini minimi da $R$ e da $Q$, per poi condurre i discenti a comprendere al fatto che il numero di percorsi minimi da $P$, risulta essere la somma del numero di percorsi minimi da $R$ ad $O$ e da $Q$ ad $O$.

**L'algoritmo ricorsivo** [lezione frontale]
Per i casi non base, vale la seguente equazione

$$\mathbf{M}(m, n) = \mathbf{M}(m-1, n) + \mathbf{M}(m, n-1) \tag{6}$$

che permette di ricondurre il problema di determinare il numero di cammini minimi da $P$, al problema di determinare il numero di cammini minimi che partono, rispettivamente, dai due punti $Q$ ed $R$, che sono più vicini alla meta di quanto lo fosse $P$ (rispetto la distanza euclidea, cioè a volo d'uccello).

I punti $P$ per cui vale l'equazione (6) si dicono **casi ricorsivi**, e l'equazione (6) si chiama **regola ricorsiva**. Questa terminologia è legata al fatto che il problema per un caso ricorsivo può essere ricondotto all'analogo problema per una nuova coppia di casi. Questi ultimi potranno essere, a loro volta, casi ricorsivi o base. Come abbiamo visto precedentemente, per i casi base la soluzione è immediata, basta applicare direttamente (5), mentre per quelli ricorsivi si ricorre nuovamente alla regola ricorsiva (6), finché, dopo un opportuno numero (è ancora da accertarsi che esso sia finito!) di applicazioni delle due regole , ci si riconduce a considerare punti che giacciono sugli assi coordinati, ovvero casi base, permettendo così la soluzione (computazionale) del problema di partenza.

Il processo di computazione sopra descritto è un **algoritmo ricorsivo**. Esso è descritto dalla coppia di regole (5-6), cioè

$$\begin{cases} \forall m, n \in \mathbb{N} \;:\; \mathbf{M}(m, 0) = \mathbf{M}(0, n) = 1 \\ \forall m, n \in \mathbb{N}_0 :\; \mathbf{M}(m, n) = \mathbf{M}(m-1, n) + \mathbf{M}(m, n-1) \end{cases}$$

che permettono di definire univocamente la funzione **M** che, per ogni punto $P(m,n)$ del primo quadrante, restituisce il numero $\mathbf{M}(m,n)$ di percorsi di minima lunghezza che congiungono il dato punto $P$ ad $O$. Nel linguaggio della matematica, una funzione come quella sopra si dice **definita per ricorsione**.

**Esercizi da svogersi in classe** [discussione]
Determinare $M(1,1)$, $M(2,1)$, $M(3,2)$, applicando l'algoritmo ricorsivo.

**Esercizi da svogersi a casa** [lavoro individuale]

- Determinare $M(2,3)$ e quindi $M(3,3)$.
- Completare la dimostrazione lasciata come esercizio.
- Siamo certi che l'algoritmo termini dopo un numero finito di passi? In tal caso, come dimostrarlo?

# 4 Scheda n. 4d: simulazione dell'algoritmo ricorsivo

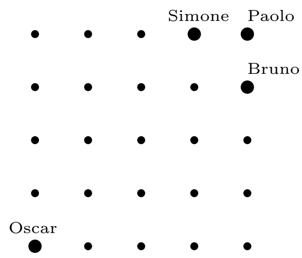
Lo scopo del gioco-simulazione che viene descritto in questa scheda è mostrare che applicando le regole dell'algoritmo ricorsivo è possibile in un tempo finito trovare il numero di percorsi minimi tra due incroci di una piccola rete stradale, rappresentata dagli stessi discenti.

**Suggerimento per il docente** La simulazione può essere realizzata in classe o in uno spazio aperto. Nella versione qui proposta, richiede 25 discenti, ma può essere adattata, in maniera ovvia, anche per gruppi classe meno consistenti. Al contrario, si sconsiglia di realizzare la simulazione per un numero di incroci superiore a 25, perché comporterebbe una durata di oltre un'ora.

**Materiali necessari**

- almeno 70 biglie raggruppate in tre o quattro cestelli;
- un cartellone e materiale di cancelleria;
- 25 blocknotes, altrettante penne e ciotole capienti (una ciotola deve poter contenere tutte le settanta bilgie);
- un orologio per misurare la durata del gioco.

**Preparazione del gioco**
In un spazio liberato dai banchi, si dispongono 25 sedie a formare una griglia di 5 righe e 5 colonne, e ciascun discente occupa un posto a sedere. Ogni discente deve avere penna e blocknotes a portata di mano. Sulla lavagna si rappresenta uno schema con nome e posizione di ciascun discente, analogo a quello riportato nella seguente figura.




Nella griglia, che qui funge da esempio, Paolo, che è seduto nell'angolo in alto a destra, rappresenta l'incrocio di partenza dei percorsi, mentre Oscar, che è seduto in basso e a sinistra, ne rappresenta la meta. Tutti gli altri discenti rappresentano i vari incroci del rettangolo di rete stradale tra Oscar e Paolo.

Il risultato finale del gioco sarà trovare il numero di percorsi minimi tra l'incrocio di partenza e quello di arrivo, utilizzando le regole dell'algoritmo ricorsivo soluzione del problema “Manhattan”.

**Le regole del gioco**

1. gli incroci si dividono in incroci base, cioè gli incroci nella riga e nella colonna di Oscar, e incroci ricorsivi, che sono tutti gli altri casi;

2. ogni rappresentante di incrocio ricorsivo può richiedere all'incrocio adiacente nella colonna a sinistra e a quello adiacente nella riga in basso, che consegnino un numero di biglie pari al rispettivo numero percorsi minimi che li congiungono alla meta. Una volta che queste due richieste sono state soddisfatte, la somma dei due gruppi di biglie, in virtù della regola ricorsiva, equivale al numero di percorsi minimi che congiungono l'incrocio considerato con la meta;

3. ogni rappresentante di incrocio base, invece ad ogni richiesta pervenuta dall'incrocio adiacente nella colonna a destra, o da quello nella righa in alto, risponde prendendo una biglia dal cestino più vicino e consegnandola all'incrocio richiedente;

4. inizialmente nessuno dei 25 rappresentanti degli incroci possiede alcuna bilgia;

5. il gioco inizia da Paolo che, in conformità con la regola 3, richiede le biglie a Simone e Bruno. Questi ultimi, per soddisfare la richiesta, dovranno instradare ulteriori richieste, e così via fino a giungere gli incroci base che, finalmente, risponderanno ad ogni richiesta consegnando una biglia;

6. ogni rappresentante di incrocio annoterà nel proprio block notes il numero di richieste che ha dovuto soddisfare per ciascun rappresentante richiedente, e il numero totale di biglie che ha consegnato ad ogni richiesta;

7. finché un incrocio non ha soddisfatto una richiesta, cioè finché non ha consegnato le biglie al richiedente, non potrà occuparsi di altre nuove richieste, che però accoderà in una lista di attesa nel blocknotes;

8. il gioco termina quando la richiesta di Paolo è stata soddisfatta, e quindi quando sono state soddisfatte tutte le richieste che essa ha generato.

**Spunti di riflessione da proporre ai discenti prima di iniziare il gioco** [intervista guidata]

- Terminerà il gioco (in un tempo finito)?

- Se effettivamente il gioco termina, quante bilgie si trova il rappresentante dell'incrocio iniziale?

**Conclusione della simulazione e raccolta dati** [lavoro di gruppo]

Al termine della simulazione, che dovrebbe durare circa un'ora, che varrebbe comunque la pena di cronometrare, verrà preparato un cartellone riassuntivo dei dati raccolti nei block notes.

Utilizzando una piccola parte del tabellone, i discenti predisporrnno due tabelle con 5 righe e 5 colonne.

Per entrambe, a ciascuna cella verrà fatto corrispondere il relativo rappresentante di incrocio, secondo la disposizione che egli aveva nel gioco.

Nella prima tabella si scriverà per ogni cella il numero di biglie che il relativo rappresentante ha consegnato ad ogni richiesta; nella cella relativa a Paolo, invece, sarà annotato il numero totale di biglie ricevute da Simone e da Bruno, cioè la risposta al problema iniziale, che risulta essere 70.

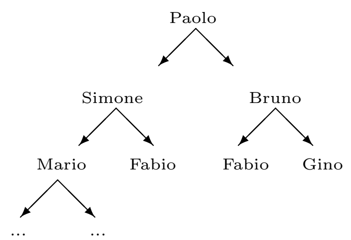
Nella seconda, invece, si scriverà il numero di richieste che ciascun rappresentante ha dovuto soddisfare, ponenedo nelal cella relativa a Paolo il numero 1.

Le tabelle risulteranno essere, nell'ordine, le seguenti

| | | | | |
|---|---|---|---|---|
| 1 | 5 | 15 | 35 | 70 |
| 1 | 4 | 10 | 20 | 35 |
| 1 | 3 | 6 | 10 | 15 |
| 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 1 | 1 | 1 |

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 4 | 4 | 3 | 2 | 1 |
| 10 | 10 | 6 | 3 | 1 |
| 20 | 20 | 10 | 4 | 1 |
| 0 | 20 | 10 | 4 | 1 |

Nella seconda parte del cartellone verrà rappresentata una parte dell'**albero di ricorsione** (non la totalità, perché altrimenti l'albero "cresce" spropositamente!), così da fornire ai discenti uno schema, pur parziale, dell'evoluzione della simulazione.




Il vertice dell'albero è l'incrocio rappresentato da Paolo e i primi due rami, come rappresentato nella figura sottostante, rappresentano le due richieste che egli pone a Simone e Bruno. A sua volta, Simone pone due richieste, una a Mario (a sinistra di Simone) e una a Fabio (sotto a Simone); anche Bruno pone una coppia di richieste, una a Fabio (sinistra di Bruno) e una a Gino (sotto a Bruno). Si procede in questo modo, fino a terminare sui rappresentanti degli incroci base, che non propagano ulteriori richieste.

**Spunti di riflessione da proporre ai discenti come successivo lavoro da svolgersi a casa**

- Leggendo la tabella dall'angolo in basso a sinistra si potrà riconoscere il triangolo di Pascal; come spiegare questa cosa in relazione alle regole dell'algoritmo?
- quante sono state, complessivamente, le richieste assolte? Dunque, quanto è stata la durata media dell'intervallo di tempo tra richiesta e consegna delle biglie?
- Dopo aver assolto alla prima richiesta, ciascun rappresentante sa esattamente (lo ha scritto nel blocknotes!) quante biglie consegnerà ad ogni successiva richiesta. Cosa cambierebbe nel gioco se i rappresentanti di ciascun incrocio, alle richieste successive alla prima, consegnassero direttamente le biglie prendendole da un proprio sacchetto, invece di propagare ulteriori richieste agli incroci adiacenti?

# 5 Scheda n. 5d: implementazione dell'algoritmo ricorsivo

In questa scheda viene presentata l'attività di laboratorio di informatica connessa all'implementazione dell'algoritmo ricorsivo soluzione del problema "Manhattan". Il linguaggio di programmazione che qui supponiamo di utilizzare è il Pascal, ma, ovviamente, qualunque linguaggio che supporti le funzioni ricorsive può essere ugualmente adoperato.

**Fase preliminare** [lezione frontale, discussione]
Inizialmente il docente proporrà alcuni semplici algoritmi ricorsivi, ad esempio, l'algoritmo per il calcolo del fattoriale di un intero $n$ ($n!$) e quello per il calcolo della potenza $n$-esima di un numero reale, ne mostrerà l'implementazione in Pascal, introducendo le *funzioni ricorsive*.

Esempi semplici di algoritmi ricorsivi sono l'algoritmo per il calcolo del fattoriale di un intero $n$ ($n!$) e quello per il calcolo della potenza $n$-esima di un reale, che possono essere realizzati in Pascal attraverso le seguenti funzioni ricorsive:

```
function fattoriale(n: integer): integer;
begin
   if (n = 0) or (n = 1) then fattoriale:=1
   else fattoriale:= n*fattoriale(n - 1)
end;

function potenza(a: real; n: integer): real;
begin
   if n = 0 then potenza:=1
   else potenza:= a*potenza(a, n - 1)
end;
```

I discenti, che hanno precedentemente ricevuto una scheda (Scheda 3s) con i listati delle funzioni ricorsive senza i commenti, dovrebbero a questo punto analizzare sintassi e semantica delle definizioni ricorsive presentate, individuando da soli i casi base e i casi ricorsivi. Terminata l'analisi dei listati, si passa alla successiva fase sperimentale.

**Fase della sperimentazione** [attività in laboratorio di informatica, lavoro di gruppo]

I discenti, divisi a gruppi di lavoro due o tre componeneti, proveranno a realizzare un programma con una delle funzioni ricorsive presentate, per avere una riprova sperimentale del funzionamento delle funzioni ricorsive.

Successivamente, i gruppi di lavoro proveranno a scrivere un programma con una funzione ricorsiva che realizzi l'algoritmo risolutivo del problema "Manhattan".

Il listato finale del programma potrà essere simile al seguente:

```
program manhattan;
var mm,nn: integer;
function M(mm, nn: integer): longint;
begin
   if (nn = 0) or (mm = 0) then M:=1 (* casi base *)
   else M:=M(mm − 1, nn)+M(mm, nn − 1)(* casi ricorsivi *)
end;
begin
   writeln('Inserisci le coordiante ( m, n) del punto P');
   write('m= ');
   readln(mm);
   write('n= ');
   readln(nn);
   writeln('Vi sono ', M( m, n), ' percorsi minimi da P ad O');
   readln
end.
```

Quando Nelle prime esecuzioni del programma, si potrà verificare la correttezza dei risultati trovati "a mano" per gli esempi proposti in classe o come lavoro per casa e per il gioco-simulazione, realizzato nella lezione precedente.

Successivamente, facendo un po' di prove per punti del tipo $P(n, n)$ via via più distanti dall'origine, si scopre che il valore $\mathbf{M}(n, n)$ cresce in modo esponenziale, e che il tempo di computazione dell'algoritmo, praticamente inavvertibile finché n è piccolo (tipicamente per valori inferiori a 8, dipendentemente dalla velocità della macchina), cresce anch'esso esponenzialmente, per arrivare a parecchie decine di secondi già per valori di $n$ attorno a 15. Per dare uan collocazione più sistematica alle precedenti osservazioni qualitative, i discenti potrebbero raccogliere i valori $\mathbf{M}(n, n)$ e i relativi tempi di esecuzione del programma, misurati con un cronometro da polso, per $n = 1, \ldots, 16$.

**Relazione dell'attività** [lavoro di gruppo]
I discenti stenderanno una sintetica relazione dell'esperienza di laboratorio, comprendente il listato del programma, un grafico dell'andamento di $\mathbf{M}(n, n)$ al crescere di $n$, e dei tempi di computazione del programma, una riflessione sul perché dell'aumento vertiginoso dei tempi di computazione, cercando di darne una possibile spiegazione.

Le relazioni dovrebbero essere consegnate al docente in tempo utile per poter essere commentate durante la lezione successiva (Lezione 6) dell'Unità di Lavoro.

**Esercizi per casa** [lavoro individuale]

Come esercizi per casa da proporre ai discenti potrebbero esserci:

- estendere la definizione della funzione potenza al caso di esponenti negativi;
- realizzare una funzione ricorsiva che realizzi $n \to (n!)!$;
- realizzare una funzione che realizzi $n \to 1 + \frac{1}{2!} + \cdots + \frac{1}{n!}$

# 6 Scheda n. 6d: il problema "Manhattan" e il Principio di Induzione

In questa scheda viene presentata la dimostrazione del fatto che l'algoritmo che si è scoperto per il problema "Manhattan" termina in un numero finito di passi qualunque sia il punto di partenza $P$.

**Discussione esercizi e relazione** [discussione]
Il docente coreggerà gli esercizi assegnati nella Lezione 5, e discuterà le relazioni sull'esperienza di laboratorio. Infine, la discussione verrà orientata al problema della terminazione dell'algoritmo, su quali basi poggi la congettura della terminazione dell'algoritmo per un qualunque punto $P$, che tanto gli esempi studiati, quanto la speriemntazione in laboratorio hanno confermato.

**Dimostrazione della terminazione dell'algoritmo** [lezione frontale]
L'algoritmo ricorsivo, è dato dalle due seguenti equazioni

$$\forall m, n \in \mathbb{N} : \mathbf{M}(0, n) = \mathbf{M}(m, 0) = 1 \tag{7}$$

$$\mathbf{M}(m, n) = \mathbf{M}(m, n-1) + \mathbf{M}(m-1, n) \tag{8}$$

Tuttavia non abbiamo provato che siamo in grado di calcolare $\mathbf{M}(m, n)$, per ogni incrocio $P \equiv (m, n) \in \mathbb{N} \times \mathbb{N}$ della nostra rete stradale, utilizzando un numero *finito* di volte le regole (7) e (8), cioè per *ricorsione finita*.

Innanzitutto, per quanto riguarda la calcolabilità in un numero finito di passi, sappiamo che

1. *per ciascun caso base, cioè per i punti $P \equiv (m, n)$ disposti sugli assi cartesiani, il problema è risolto con una sola applicazione della regola (7), che pone $\mathbf{M}(m, n) = 1$.*

2. *per i casi ricorsivi, in virtù della regola (8), la calcolabilità per ricorsione finita di $\mathbf{M}(m, n)$ deriva dalla calcolabilità per ricorsione finita di $\mathbf{M}(m-1, n)$ e $\mathbf{M}(m, n-1)$.*

Lo schema di ricorsione che abbiamo delineato nasconde una forte parentela con il Principio di Induzione:

**Principio di induzione** *Data la famiglia di predicati $\mathbf{P}_h$ al variare di $h \in \mathbb{N}$, se sussistono i seguenti due fatti*

1. *$\mathbf{P}_0$ vero (base dell'induzione)*

2. *per ogni $h \in \mathbb{N}$ si ha che $\mathbf{P}_h \Rightarrow \mathbf{P}_{h+1}$*

*allora sia ha che $\mathbf{P}_h$ è vero per ogni $h \in \mathbb{N}$.*

Questo legame non è solo apparente, ma come vedremo nel successivo teorema, su di esso si può fondare la dimostrazione della calcolabilità per ricorsione finita della funzione $\mathbf{M}$, che viene di seguito presentata.

**Teorema 1** *La funzione* $M : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *che risolve il* problema di Manhattan *è calcolabile mediante risorsione finita.*

**Dim..** Innanzitutto, per ogni $h$ in $\mathbb{N}$, definiamo il seguente sottoinsieme di $\mathbb{N} \times \mathbb{N}$

$$\mathbf{C}_h \triangleq \{(i, j) \in \mathbb{N} \times \mathbb{N} |\ ij \leq h\}$$

E' immediato costatare che

- per ogni $h \in \mathbb{N}$ si ha $\mathbf{C}_h \subset \mathbf{C}_{h+1}$,
- l'unione degli insiemi $\mathbf{C}_h$, al variare di $h \in \mathbb{N}$, è $\mathbb{N} \times \mathbb{N}$.

Consideriamo ora, al variare di $h$ in $\mathbb{N}$, il seguente predicato:

$$\mathbf{P}_h \triangleq \{\mathbf{M}(m, n) \text{ è calcolabile mediante ricorsione finita per ogni coppia } (m, n) \in \mathbf{C}_h\}$$

E' evidente che sussiste $\mathbf{P}_0$, dato che i punti di $\mathbf{C}_0$ sono esattamente i casi base, per i quali $\mathbf{M}(m, n)$ vale 1 applicando una volta la regola (7).

Se assumiamo $\mathbf{P}_0$ come base dell'induzione sull'indice $h$, la tesi discende dal Principio di Induzione una volta che avremo provato che

$$\forall h \in N : \mathbf{P}_h \to \mathbf{P}_{h+1}$$

Sia dunque $h$ arbitrario naturale, dobbiamo mostrare che stante l'ipotesi induttiva $\mathbf{P}_h$, per una qualunque coppia $(m, n)$ in $\mathbf{C}_{h+1}$, il valore $\mathbf{M}(m, n)$ può essere calcolato per ricorsione finita mediante le regole (8) e (8).

Dato che $\mathbf{P}_0$ è vera, ci si può limitare ai punti $P \equiv (m, n)$ non appartenenti a $\mathbf{C}_0$, cioè tali per cui $m, n \geq 1$.

In virtù della relazione di ricorsione (8), se proviamo che che entrambi $\mathbf{M}(m, n-1)$ e $\mathbf{M}(m-1, n)$ sono calcolabili per ricorsione finita, ne discenderà $\mathbf{P}_{h+1}$, e quindi la tesi del teorema.

Verifichiamo solo la calcolabilità di $\mathbf{M}(m, n-1)$, dato che per $\mathbf{M}(m-1, n)$ basta procedere con un analogo argomento.

In virtù dell'ipotesi induttiva $\mathbf{P}_h$, per mostrare la calcolabilità di $\mathbf{M}(m, n-1)$ basterà provare che $(m, n-1)$ è punto in $\mathbf{C}_h$, cioè che $m(n-1) \leq h$. Quest'ultimo fatto si può provare come segue. Essendo $(m, n) \in \mathbf{C}_{h+1}$, si ha che $mn \leq h + 1$, dunque

$$m(n-1) = mn - m \leq h - (m-1) \tag{9}$$

Ora si rammenti che abbiamo supposto $m \geq 1$, dunque $h - (m-1) \leq h$ e, tenuto conto di (9), ne discende immediatamente che $(m-1)n \leq h$, come volevasi dimostrare. ■

# 7 Scheda n. 7d: soluzione combinatoria dele problema "Manhattan"

In questa scheda viene presentata una soluzione alternativa del problema "Manhattan', ottenibile in modo elegante e immediato con metodi combinatori. Prerequisito di questa parte della lezione è quindi di alcuni elementi di *Calcolo Combinatorio*, che, nella fattispecie, sono le permutazioni semplici o le permutazioni con ripetizione. La metodologia didattica consigliata è la lezione frontale, abbinata a momenti di discussione delle congetture proposte, e di successive dimostrazioni formali.

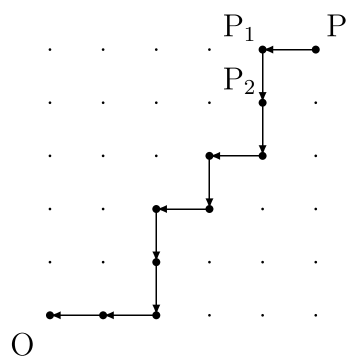
**Dalla caratterizzazione dei cammini minimi alle permutazioni** [discussione, lezione frontale]
La caratterizzazione derivata nella Lezione 2 (vedi Scheda 2d), permette di associare ad un cammino minimo da P ad O, una sequenza di istruzioni di movimento di una lunghezza $L$ lungo le sole direzioni Ovest (O) e Sud (S). Più precisamente, se P ha coordinate $(m, n)$, allora

1. i cammini minimi da $P$ ad $O$ hanno lunghezza $n + m$;
2. i cammini minimi da $P$ ad $O$ sono tutti e soli quelli le cui sequenze di movimento $\Sigma$ che hanno $m$ istruzioni di movimento in direzione Ovest e $n$ istruzioni di movimento in direzione Sud.

Ad esempio, la sequenza

$$\Sigma = (O,\ S,\ S,\ O,\ S,\ O,\ S,\ S,\ O,\ O)$$

rappresenta il cammino mostrato nella seguente figura.



I cammini minimi sono tanti quanti i possibili allineamenti distinti di $m$ simboli $O$ e $n$ simboli $S$, dunque determinare $\mathbf{M}(m, n)$ può quindi essere interpretato come un problema di tipo combinatorio.
La soluzione di questo problema è immediata facendo ricorso alle permutazioni con ripetizione, ma può tranquillamente essere ottenuta anche utilizzando le permutazion isemplici; quest'ultima è la strada qui proposta.
E' noto che $P_{n+m} = (n+m)!$ sono i modi distinti con cui si può permutare l'ordine degli $n+m$ oggetti, tuttavia, essendoci degli oggetti che si ripetono, occorre sottrarre da questo numero le ripetizioni si sequenze identiche. Per fare questo, si considera uno qualunque di questi $(m+n)!$ ordinamenti dei simboli $O$ e $S$, e si osserva che sono $P_n = n!$ le permutazioni che scambiano di posto gli $n$ simboli $S$ lasciando fermi i simboli $O$, mentre sono $P_m = m!$ le permutazioni che scambiano di posto gli $m$ simboli $O$ lasciando fermi i simboli $S$.

Dunque sono ben $n!\,m!$ le permutazioni che non alterano la data sequenza di istruzioni di movimento. Di conseguenza, il numero di sequenze distinte di movimento è dato dal comando iterativo

$$\mathbf{M}(m, n) = \frac{(m+n)!}{m!\,n!}.$$

### Suggerimento per il docente

Un'altra strada per risolvere il problema senza far ricorso né a ricorsione né a comandi iterativi, può essere percorsa partendo da un'osservazione fatta nella Lezione 4 (scheda 4d), cioè che la tabella dei valori di $\mathbf{M}(m, n)$, se letta nella direzione opportuna, coincide con il triangolo di Pascal. Osservato ciò, si può facilemente ideare un algoritmo iterativo che, ricalcando quello con cui si costruisce il triandolo di Pascal, riempia una matrice di ordine $(m+1) \times (n+1)$ con i valori assunti dalla funzione $\mathbf{M}$ nei punti $(0, 0)$, $(0, 1), \ldots, (m, n)$.

# 8 Scheda n. 8d: altri problemi di tipo ricorsivo

In questa scheda vengono presentati 4 nuovi problemi con una traccia delle loro soluzioni ottenute mediante un approccio di tipo ricorsivo. Questi sono: un problema combinatorio, un problema di tassellazione, un problema di ricerca dicotomica ed infine il celebre problema della Torre di Hanoi.

### Suggerimento per il docente

Il problema di ricerca dicotomica, qui titolato *Una moneta falsa da individuare*, può anche essere assegnato come esercizio per casa, essendo il più facile dei quattro presentati.

La presenza di ben quattro problemi, permette di mediare tra metodologie didattiche differenti nel corso della lezione. Tuttavia, le soluzioni dei problemi, pur nella loro estrema semplicità, non sono affatto banali e richiedono pazienza per essere comprese ed assimilate. La scansione temporale della lezioen impone limiti abbastanza stretti, pere cui bisogna attentamente modulare i momenti in cui, abbandonata per un tratto la lezione frontale, si avvia una discussione con i discenti.

### Il problema delle $n$ sedie

Data una fila di $n$ posti a sedere, in quanti modi diversi si possono assegnare i posti a uomini e donne di modo che le donne non siedano mai in posti contigui?

**Soluzione** Detto $a_n$ il numero di modi di organizzare gli $n$ posti a sedere, si verifica facilmente che $a_1 = 2$ e $a_2 = 3$; questi sono i casi base. Supposto che sia $n \geq 2$, è possibile ricondurre il problema di determinare $a_{n+1}$ ai casi $n$ e $n-1$. Data la fila di $n+1$ posti, consideriamo i primi $n$ posti; delle $a_n$ configuarazioni per questi $n$ posti ne avremo $F_n$ con l'$n$-esimo posto oocupato da una femmina e $M_n$ configurazioni con l'$n$-esimo posto occupato da un maschio. Dunque, per le $F_n$ configurazioni, il posto $n+1$-esimo dovrà essere occupato necessariamente da un maschio, mentre per le altre $M_n$ configurazioni, potrà esserlo tanto da un maschio quanto da una femmina. Quindi, risulta $a_{n+1} = F_n + 2M_n$ e, ricordando che $F_n + M_n = a_n$, si potrà scrivere $a_{n+1} = a_n + M_n$. A questo punto è facile riconoscere che $M_n = a_{n-1}$, infatti ad ogniuna delle $a_{n-1}$ configurazione per i primi $n-1$, corrisponde esttamente una configuarazione per i primi $n$ posti ottenuta aggiungendo alla prima un posto per un maschio.

Abbiamo quindi derivato un algoritmo ricorsivo che risolve il problema in esame. Le sue equazioni sono

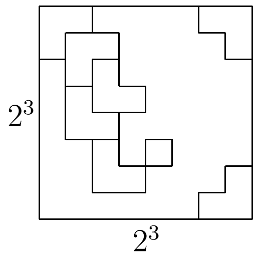
$$\begin{cases} a_1 = 2, \; a_2 = 3, \\ \forall n \geq 2: \; a_{n+1} = a_n + a_{n-1}. \end{cases}$$

La successione $\{a_n\}_{n \geq 1}$, definita per ricorsione dalle regole appena specificate, è nota anche col nome di serie di Fibonacci.

**Un problema per il piastrellista**

Immagina di voler coprire il pavimento di una stanza quadrata di lato $2^k$ unità, avendo a disposizione solo una piastrella quadrata di lato 1 unità e quante si voglia piastrelle a forma di L, con i due lati maggiori lunghi 2 unità e i restanti lati 1 unità, come mostrato nella seguente figura (caso $k = 3$).

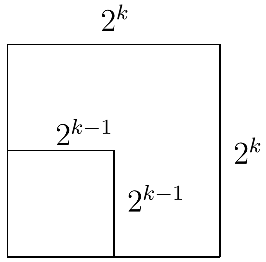
Le piastrelle possono essere disposte sul pavimento in qualunque orientazione e non possono essere tagliate, inoltre non sono ammesse sovrapposizioni tra piastrelle o piastrelle che sbordano dal quadrato.
Proponi una soluzione che possa andar bene per ogni valore positivo dell'intero $k$.



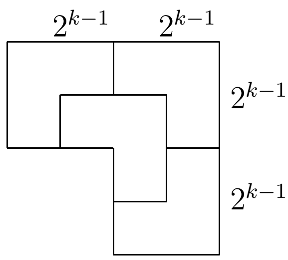

**Soluzione** Il problema può essere affrontato con un approccio ricorsivo. Dato il problema $P_k$, con $k \geq 1$, ovvero il caso del pavimento con lato $2^k$, si divide tale paviemento in due regioni, come riportato nella seguente figura.

Consideriamo ora i due sottoproblemi

- $A_k$: tassellare la regione quadrata di lato $2^{k-1}$ con una mattonella quadrata e un numero opportuno di mattonelle a $L$,
- $B_k$ : tassellare la regione a forma di L, con lati maggiori $2^k$ e lati minori $2^{k-1}$, con sole mattonelle a forma di $L$,




Se si è in grado di risolvere $A_k$ e $B_k$ allora il caso $k$-esimo è risolto. Il problema $A_k$ corrisponde esattamente al problema $P_{k-1}$; mentre, il problema $B_k$, può essere ricondotto $B_{k-1}$ con la suddivisione della regione a forma di L riportata nella figura a fianco.
In questo modo il problema $P_k$ viene ricondotto al problema $P_{k-1}$. L'algoritmo ricorsivo è quindi completo una volta che è risolto il caso base, cioè il problema $P_1$; quest'ultimo è immediatamente risolto con una mattonella quadrata e una a forma di L.



**La Torre di Hanoi**

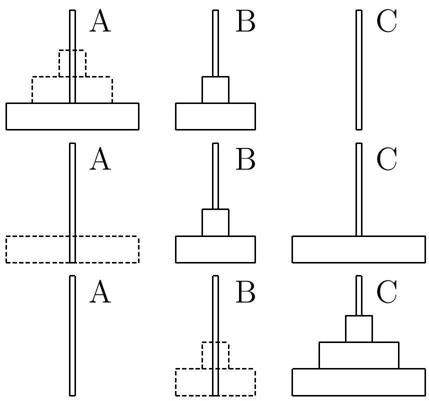
Su una piattaforma vi sono tre pioli, che chiamiamo A, B, C. Sul piolo A sono infilati $n$ dischi di dimensione crescente a formare una torre. Prova ad individuare una strategia per portare la torre dal piolo A al piolo C, spostando un solo disco alla volta e in modo che i dischi di diametro maggiore si trovino sempre sotto a dischi di diametro minore; sempre rispettando questa regola, per spostare i dischi, si possono utilizzare tutti e tre i perni.

**Soluzione** Diciamo $P_n$ il problema di spostare la torre di $n$ dischi. Per $n = 1$ il problema si risolve con una sola mossa spostando l'unico disco da A a C. Per $n = 2$, invece, basta spostare il disco 1 sul piolo B, il diso 2 sul piolo C, ed infine il disco due dal piolo B al piolo C. Supponiamo ora $n \geq 3$, e consideriamo la sequenza di problemi:

$\mathrm{A}_n$ : spostare il gruppo degli $n-1$ dischi superiori della torre dal piolo A al piolo B;

$\mathrm{B}_n$ : spostare l'$n$-esimo disco da A a C;

$\mathrm{C}_n$ : il gruppo degli $n - 1$ dischi dal piolo B al piolo C.




Una volta risolto $A_n$, spostare il disco di diametro maggiore al piolo C, equivale al problema $P_1$, che può essere risolto in una mossa. D'altra parte, risolvere il problema $A_n$ equivale a risolvere il problema $B_n$, ed entrambi equivalgono a risolvere il problema $P_{n-1}$. Dunque, il problema di spostare una torre di $n$ dischi, può essere ricondotto al problema di spostare una torre di $n - 1$ dischi con il procedimento sopra descritto. Si è dunque trovata una soluzione ricorsiva del problema in esame, il cui caso base è quello di una torre con 1 disco.

Si noti che il numero di chiamate ricorsive per il problema $P_n$ è $2^n - 1$, dunque cresce esponenzialmente al crescere del numero $n$ di dischi.

### Una moneta falsa da individuare

Immagina di avere $2^k$ monete identiche all'apparenza, di cui $2^k - 1$ sono d'oro massiccio, mentre 1 è falsa, essendo placcata d'oro. Le monete d'oro massiccio hanno tutte la stessa massa, mentre quella falsa ha una massa leggermente inferiore. Supponendo di aver a disposizione una bilancia a due piatti, con la quale puoi verificare se le monete sul piatto sinistro pesano di più di quelle sul piatto destro, prova a ideare una strategia che consenta di identificare la moneta falsa con un numero minimo di pesate. Durante la singola pesata non è possibile né aggiungere né togliere monete dai piatti della bilancia.

**Soluzione** Il problema può essere risolto con una semplice strategia ricorsiva di ricerca binaria. Se $k = 1$, cioè si hanno 2 monete di cui 1 è falsa, una singola pesata è necessaria e sufficiente per determinare la moneta falsa; questo è il caso base. Se consideriamo il caso di $2^k$ monete, con $k \geq 2$, posssiamo dividere le monete in due gruppi di $2^{k-1}$ monete ciascuno, e con una singola pesata decidere in quale gruppo c'è la moneta falsa; tra i due, è chiaramente il gruppo che pesa di meno. A questo punto il problema si riconduce al problema di trovare la moneta falsa nel gruppo di $2^{k-1}$ monete, cioè al caso $k - 1$-esimo.

### Esercizi da assegnare per casa

- Risolvere il problema *Una moneta falsa da individuare* con un approccio ricorsivo; cosa cambia se la moneta falsa pesa di più di quelle d'oro? E nel caso in cui non si conosca affatto il peso della moneta falsa?
- Provare a risolvere sperimentalemnte il problema della Torre di Hanoi con 4 dischi, applicando le regole dell'algoritmo ricorsivo trovato.

# 9 Scheda n. 9d: la valutazione finale

In questa scheda vengono presentati alcuni spunti per preparare un test di valutazione da somministrare ai discenti al termine delle attività dell'Unità di Lavoro. La funzione del test è quella di misurare competenze e abilità operative acquisite dai discenti relativamente ad argomenti ed esperienze affrontati nell'Unità.

La traccia di verifica qui presentata, comprende anche un'attività in laboratorio d'informatica, per cui è stata strutturata prevedendo due fasi temporalmente e spazialmente distinte.

**Prima fase: test scritto individuale** [ durata 1 ora]

La prima fase consiste in una verifica scritta individuale, della durata di un'ora, mirata ad evidenziare le competenze che ciascun discente ha acquisito. Di seguito sono proposti tre tipologie di quesiti che potrebbero essere inclusi in nella verifica scritta.

**Primo quesito**

Il primo questo potrebbe essere la comprensione e analisi di un dato algoritmo ricorsivo, formalizzato in Pascal. Ad esempio, data la funzione

```
function T(parola : string): integer;
var   lunghezza: integer;
      parolatroncata: string;
begin
   lunghezza: = length(parola);
   if lunghezza=0 then T := 0 (* caso base *)
   else
      begin
         paolatroncata:=copy(parola, 1, lunghezza-1);
         T=T(parolatroncata)+ 2 (* caso ricorsivo *)
      end
end;
```

si può chiedere quale è il risultato di **T**('Ape regina'), che risulta il doppio del numero di caratteri della stringa in argomento.

**Secondo quesito**

Il secondo quesito potrebbe focalizzarsi sul problema del raggiungimento o no delle condizioni di terminazione di un dato algoritmo ricorsivo. Ad esempio, dato il seguente algoritmo ricorsivo

$$\begin{cases} \text{se } 1 > x \geq 0 \text{ allora } \mathbf{B}(\text{x})\text{=0}, \\ \text{altrimenti } \mathbf{B}(x) = 1 + \mathbf{B}(x-1), \end{cases}$$

si possono porre ai discenti le seguenti questioni:

- individua quali sono i casi base dell'algoritmo ricorsivo e quali quelli ricorsivi;
- per quali valori dell'argomento reale $x$, il valore $\mathbf{B}(x)$ viene computato in un numero finito di passi?
- in tal caso, cosa rappresenta il valore $\mathbf{B}(x)$?

La risposta, come è facile intuire, è che l'algoritmo termina solo per valori dell'argomento $x$ maggiori di $-1$; in tali casi, il valore $\mathbf{B}(x)$ risulta essere il più grande intero maggiore o uguale a $x$.

**Terzo quesito**

Il terzo quesito potrebbe essere la realizzazione di un algoritmo che permetta di computare l'$n$-esimo termine di una data successione di numeri interi, e la sua formazlizzazione nel linguaggio Pascal. Ad esempio, nella successione

$$1, 2, 3, 6, 11, 20, 37, 68, \ldots ...$$

ogni nuovo termine (dal quarto in poi) si ottiene come somma dei tre precedenti. Per cui, l'algoritmo ricorsivo che risolve il problema è descritto da

$$\begin{cases} \mathbf{C}(1) = 1,\ \mathbf{C}(2) = 2,\ \mathbf{C}(3) = 3; \\ \forall n \geq 3 :\ \mathbf{C}(n+1) = \mathbf{C}(n) + \mathbf{C}(n-1) + \mathbf{C}(n-2). \end{cases}$$

Una sua possibile formalizzazione nel linguaggio Pascal è la seguente funzione ricorsiva

```
function C(n: integer): longint;
begin
   case n of
      1: C:=1; (* caso base *)
      2: C:=2; (* caso base *)
      3: C:=3; (* caso base *)
      else C:=C(n - 1)+C(n - 2)+C(n - 3) (* casi ricorsivi *)
   end
end;
```

**Seconda fase: lavoro di guppo in laboratorio** [ durata 1 ora]

La seconda fase si svolge in laboratorio di informatica. Lì, gruppetti di due o tre discenti, potranno implementare gli algoritmi ricorsivi proposti nella prima fase della verifica, studiandone empiricamente il funzionamento. Entro un termine prestabilito, ciascun gruppo stilerà una relazione dell'esperienza in laboratorio, contenente i listati dei programmi compilati e illustrante i risultati ottenuti con il calcolatore. Questa relazione andrà poi ad integrare il test della prima fase di verifica.

**Suggerimento per il docente**

A volte, un piccolo errore nella stesura del listato di un algoritmo ricorsivo può comportare l'esecuzione di un loop infinito, con l'irrimediabile blocco del compilatore. A parte gli errori, questa situazione può anche verificarsi per taluni valori dei dati iniziali, come nel caso dell'algoritmo del secondo quesito. Per ovviare a questo inconveniente, il docente dovrebbe intervenire preventivamente consigliando ai discenti di stabilire un numero massimo di chiamate per le funzioni ricorsive, magari utilizzando una variabile globale come contatore e una qualche istruzione di controllo.

Ad esempio, un programma che realizzi l'algoritmo del questito due, limitando a 10000 le chiamate ricorsive, potrebbe essere il seguente:

```
program manhattan;
var    x: real;
       risultato, chiamate: integer;
const maxric=10000;
function B(x: real): integer;
begin
   chiamate:=chiamate+1;
   if chiamate < maxric then
      begin
         if (x> −1) and (x<= 0) then B :=0 (* caso base *)
         else B:=1+B(x-1) (* caso ricorsivo *)
      end
   else B=0 (* valore di uscita convenzionale *)
end;
begin
   chiamate:=0;
   writeln('Inserisci l'argomento iniziale');
   write('x= ');
   readln(x);
   risultato:=B(x);
   if chiamate < maxric then writeln('Il risultato è ', risultato)
   else writeln('Numero massimo di ricorsioni raggiunto!');
   readln
end.
```