Problema delle 8 regine

20/12/2022

Studente: Emanuele Izzo *Matricola: 0307385*

Indice

1 Gli scacchi

Gli scacchi sono un gioco di strategia che si svolge su una tavola quadrata detta scacchiera, formata da 64 caselle (o ”case”) di due colori alternati, sulla quale ogni giocatore dispone di 16 pezzi (bianchi o neri, i quali designano i due sfidanti[1]): un re, una donna o regina, due alfieri, due cavalli, due torri e otto pedoni; ogni casella può essere occupata da un solo pezzo, che può catturare o ”mangiare” il pezzo avversario andando a occuparne la casella; obiettivo del gioco è dare scacco matto, ovvero minacciare la cattura del re avversario in modo tale che l'altro giocatore non abbia mosse legali. Ciascun pezzo degli scacchi si muove con precise modalità. Nessun pezzo può andare a occupare una casa in cui è presente un altro pezzo dello stesso schieramento; può invece muoversi su una casa occupata da un pezzo avversario, effettuando in tal caso una ”cattura”, cioè eliminando (”mangiando”) dalla scacchiera il pezzo nemico e prendendo il suo posto. Si dice che un pezzo ”attacca” o ”minaccia” una casa se esso può muoversi su di essa. I movimenti che possono effettuare le pedine sono i seguenti:

- L'alfiere può muoversi su una qualunque casa della stessa diagonale rispetto a quella in cui si trova, purché per raggiungerla non debba attraversare case occupate da pezzi (amici o avversari) e purché la casa d'arrivo non sia occupata da un pezzo amico. Ciascun alfiere non cambia mai il colore delle case su cui si muove: per questo i giocatori parlano di alfieri ”campochiaro” o ”camposcuro” a seconda del colore delle case in cui si trovano.

- La torre può muoversi su una qualunque casa della stessa traversa o della stessa colonna rispetto a quella in cui si trova, purché per raggiungerla non debba attraversare case occupate da pezzi (amici o avversari) e purché la casa d'arrivo non sia occupata da un pezzo amico.

[1] Bisogna appuntare che gli scacchi non sono necessariamente bianchi e neri; a decidere il colore attribuito a ogni giocatore è la disposizione iniziale dei pezzi (il Bianco ha la donna alla sinistra del re, il Nero alla sua destra), la quale determina a chi spetterà la prima mossa (per convenzione, apre il Bianco).

- La donna è il pezzo più potente di tutti e combina le mosse dell'alfiere con quelle della torre, potendo quindi muoversi su tutte le case della stessa traversa, della stessa colonna o della stessa diagonale della casa su cui si trova.

- Il cavallo può muoversi su una delle case a lui più vicine che non appartengono alla traversa, alla colonna e alle diagonali passanti per la sua casa di partenza. Un cavallo al centro della scacchiera ha a disposizione otto case ("rosa di cavallo") verso le quali muoversi, mentre se si trova al bordo la sua mobilità è ridotta a quattro o tre case e se si trova in un angolo a due. Il movimento del cavallo può essere immaginato come la somma di uno spostamento orizzontale di una casa e di uno verticale di due o viceversa, descrivendo una "L". Il cavallo è inoltre l'unico pezzo che, nei suoi movimenti, può attraversare anche caselle già occupate da altri pezzi: si dice che può "saltare". Si noti che a ogni mossa il cavallo cambia il colore della casa in cui si trova.

- Il pedone segue regole di movimento leggermente più complesse:

  - Alla sua prima mossa, ossia quando parte dalla posizione iniziale, ciascun pedone può muovere di una oppure due case in avanti, a scelta del giocatore, a patto che la casa di destinazione ed eventualmente la casa saltata siano libere. Nelle sue mosse successive il pedone può avanzare solo di una casa per mossa, a patto che questa sia libera. A differenza degli altri pezzi, non può muovere all'indietro.
  - Il pedone è l'unico pezzo che cattura in maniera differente da come muove. Può catturare un pezzo nemico solo se si trova su una delle due case poste diagonalmente in avanti rispetto alla sua casa di partenza, ma non può né muovere in tali case se esse sono libere né catturare i pezzi che si trovano nelle case che ha di fronte.
  - Quando, eseguendo la sua prima mossa di due case in avanti, il pedone viene a trovarsi di fianco a un pedone avversario, quest'ultimo può alla mossa successiva catturarlo *en passant*, come se il primo fosse avanzato di una sola casa. La presa en passant può essere eseguita solo come mossa successiva all'avanzamento del pedone avversario di due case.
  - Se un pedone riesce ad avanzare fino all'ottava traversa, viene promosso, ossia assume il ruolo e le capacità di movimento di un altro pezzo dello stesso colore (donna, torre, alfiere o cavallo) a scelta del giocatore, indipendentemente dai pezzi già presenti sulla scacchiera. In questo modo è dunque possibile avere un numero di esemplari di un certo pezzo maggiore rispetto a quello iniziale. L'effetto è immediato, ad esempio si può dare scacco o scacco matto con una promozione se il re avversario è nel raggio di azione del nuovo pezzo.

- Il re si può muovere in una delle case adiacenti (anche diagonalmente) a quella occupata, purché questa non sia controllata da un pezzo avversario.

Il re è l'unico pezzo che non viene mai catturato, ma solo minacciato. Quando il re di uno dei due giocatori è minacciato, trovandosi sulla traiettoria di un pezzo nemico, si dice che è "sotto scacco" e non è consentita alcuna mossa che lasci il proprio re in tale condizione. Deve quindi essere effettuata una mossa che elimini la minaccia in uno dei seguenti modi:

- Muovere il re in una delle case adiacenti, a patto che questa non sia sotto il controllo di un pezzo avversario;
- Catturare, con il re o con un altro pezzo, il pezzo avversario che si trova sulla traiettoria del re e dà origine allo scacco;
- Nel caso di minaccia da parte di donna, torre o alfiere non adiacenti al re sotto attacco, frapporre tra quest'ultimo e il pezzo che minaccia scacco un qualunque pezzo o pedone, in modo che sia quest'ultimo a essere minacciato invece del re.

Se nessuna delle mosse che il giocatore può effettuare è in grado di liberare il re dallo scacco, si tratta di scacco matto e la partita termina con la vittoria dell'avversario. Se invece il re non si trova sotto scacco ma non è possibile effettuare alcuna mossa legale (ad esempio se si ha solo il re in gioco e questo non è sotto scacco, ma tutte le case libere a esso adiacenti sono minacciate), si tratta di stallo e la partita termina con un risultato di parità, non potendo il giocatore che si trova in questa condizione muovere senza contravvenire a qualche regola del gioco.

2 Definizione del problema

Il rompicapo delle 8 regine è un problema proposto nel 1848 dallo scacchista tedesco Max Bezzei. Il problema è definito nel seguente modo:

Definizione 2.1. Data una scacchiera $8 \cdot 8$, posizionare 8 regine in modo tale che nessuna delle regine possa mangiarne altre tramite una sola mossa.

È possibile definire una versione generalizzata del problema per scacchiere di varia grandezza:

Definizione 2.2. Sia $k \geq 4$, data una scacchiera $k \cdot k$, posizionare k regine in modo tale che nessuna delle regine possa mangiarne altre tramite una sola mossa.

Possiamo definire tale problema in un altro modo ragionando in questo modo:

- Una scacchiera di dimensione $k \cdot k$ può essere vista come una griglia $\mathbb{S} := [1, k]^2$.
- Utilizzando tale rappresentazione, la posizione di ogni pezzo può essere rappresentata tramite delle coordinate $(x, y) \in \mathbb{S}$.

Per comodità non andiamo a considerare come distinguere univocamente i vari pezzi sulla scacchiera, poiché andremo a lavorare solo con le regine. A prescindere da ciò, possiamo ridefinire il problema come segue:

Definizione 2.3. Sia $k \geq 4$, data una griglia $S := [1, k]^2$, posizionare k punti $q_i = (x_i, y_i)$, con $i \in [1, k]$, in modo che, per ogni coppia $i, j \in [1, k]$ tale che $i \neq j$, abbiamo che:

- $x_i \neq x_j$.
- $y_i \neq y_j$.
- Non esiste alcun $\delta \in [1, k]$ tale che $x_i = x_j + \delta$ e $y_i = y_j + \delta$.

Quest'ultima definizione del problema ci sarà utile nel definire un sistema per generare tutte le possibili soluzioni. Prima di procedere è giusto spendere due parole sul perché $k \geq 4$. Prendendo $k < 4$, abbiamo solo 3 possibili casi:

- Per $k = 1$ il problema è risolvibile ma banale.
- Per $k = 2$ e $k = 3$ il problema non è risolvibile (a prescindere come posizioniamo le regine, saranno in grado di mangiarsi a vicenda con una sola mossa).

3 Implementazione

Partiamo dal valore GRANDEZZA che definisce sia la grandezza della scacchiera che il numero di regine con cui lavoriamo.

```
GRANDEZZA = 8
```

A questo punto possiamo iniziare a definire le varie funzioni di utility che andremo ad utilizzare:

```
# controlloCol(q_i, q_j) => boolean
# Parametri:
# - q_i -> Posizione della regina i (é un'array composto da 2 elementi)
# - q_j -> Posizione della regina j (é un'array composto da 2 elementi)
# Output:
# - Restituisce vero se le due regine non sono sulla stessa colonna
# - Restituisce falso altrimenti
def controlloCol(q_i, q_j):
  if q_i[1] == q_j[1]:
    return False
  return True

# controlloRiga(q_i, q_j) => boolean
# Parametri:
# - q_i -> Posizione della regina i (é un'array composto da 2 elementi)
# - q_j -> Posizione della regina j (é un'array composto da 2 elementi)
# Output:
# - Restituisce vero se le due regine non sono sulla stessa riga
# - Restituisce falso altrimenti
def controlloRiga(q_i, q_j):
  if q_i[0] == q_j[0]:
    return False
  return True

# controlloDiag(q_i, q_j) => boolean
# Parametri:
# - q_i -> Posizione della regina i (é un'array composto da 2 elementi)
# - q_j -> Posizione della regina j (é un'array composto da 2 elementi)
# Output:
# - Restituisce vero se le due regine non sono sulla stessa diagonale
```

```
# - Restituisce falso altrimenti
def controlloDiag(q_i, q_j):
  k = abs(q_i[0]-q_j[0])
  h = abs(q_i[1]-q_j[1])
  if k == h:
    return False
  return True

# controlloSoluzione(Q) => boolean
# Parametri:
# - Q -> Insieme delle posizioni delle regine
#        (é una tabella dove ogni riga rappresenta una regina,
#        mentre ogni colonna rappresenta una delle due coordinate)
# Output:
# - Restituisce vero se la soluzione passata é valida
# - Restituisce falso altrimenti
def controlloSoluzione(Q):
  for q_i in Q:
    for q_j in Q:
      if q_i != q_j:
        if not controlloCol(q_i, q_j): return False
        if not controlloRiga(q_i, q_j): return False
        if not controlloDiag(q_i, q_j): return False
  return True

# copiaSoluzione(Q, S)
# Parametri:
# - Q -> Insieme delle posizioni delle regine
#        (é una tabella dove ogni riga rappresenta una regina,
#        mentre ogni colonna rappresenta una delle due coordinate)
# - S -> Insieme delle soluzioni corrette
#        (é un'array tridimensionale definito in questo modo:
#        la prima dimensione definisce la soluzione considerata,
#        la seconda dimensione la regina della soluzione considerata,
#        la terza dimensione le coordinate della regina della soluzione
    considerata)
# Procedimento:
# - Realizza una copia della soluzione Q in coda ad S
def copiaSoluzione(Q, S):
  S.append([[0, 0] for i in range(GRANDEZZA)])
  i = len(S)-1
  for j, q in enumerate(Q):
    S[i][j][0], S[i][j][1] = q[0], q[1]

# stampa(Q)
# Parametri:
# - Q -> Insieme delle posizioni delle regine
```

```
#         (é una tabella dove ogni riga rappresenta una regina,
#        mentre ogni colonna rappresenta una delle due coordinate)
# Procedimento:
# - Stampa la soluzione Q disegnando la scacchiera
def stampa(Q):
  C = [[" " for i in range(GRANDEZZA)] for j in range(GRANDEZZA)]
  i = 0
  for q in Q:
    C[q[0]][q[1]] = "Q"
  for j in range(GRANDEZZA*2+1):
    if j % 2 == 0:
      for k in range(GRANDEZZA):
        print("+-", end = "")
      print("+")
    else:
      for k in range(GRANDEZZA):
        print(f"|{C[i][k]}", end = "")
      print("|")
      i += 1
```

A questo punto non rimane altro che definire la nostra funzione che va a trovare tutte le possibili soluzioni. Il meccanismo dietro è il seguente:

- Quando viene chiamata `trovaSoluzioni()`, inserisco nella lista `Q` la posizione della regina sulla prima riga (a cui daremo indice 0), e chiamo ricorsivamente la funzione `generaSoluzioni(Q, 1, S)`.
- Quando viene chiamata `generaSoluzioni(Q, i, S)`, inserisco in coda la posizione della regina sulla riga `i+1` (ricordiamo che questa è rappresentata dall'indice `i`).
- Una volta che `i` è uguale ad 8, si effettua il controllo della soluzione, e se è valida la si inserisce in `S`.
- Per generare tutte le possibili posizioni delle regine usiamo la funzione `Q.pop()`: questa funzione elimina l'ultimo elemento in coda, andando a rimouvere quindi l'ultima regina inserita. Questo ci permette di usare una sola variabile per generare tutte le possibili soluzioni del rompicapo.

```
def generaSoluzioni(Q, i, S):
  if i == GRANDEZZA:
    if controlloSoluzione(Q):
      copiaSoluzione(Q, S)
  else:
    for j in range(GRANDEZZA):
      Q.append([i, j])
      if controlloSoluzione(Q):
        generaSoluzioni(Q, i+1, S)
      Q.pop()
```

```
11
12 def trovaSoluzioni():
13   Q = []
14   S = []
15   for j in range(GRANDEZZA):
16     Q.append([0, j])
17     generaSoluzioni(Q, 1, S)
18     Q.pop()
19   return S
```

In questo modo S conterrà tutte le possibili soluzioni del rompicapo delle 8 regine. Per effettaure la stampa di tutte le soluzioni è sufficiente scrivere:

```
1 for i, s in enumerate(S):
2   print(f"Soluzione {i+1}:")
3   stampa(s)
4   print("")
```

Di seguito è riportata la stampa di una sola delle soluzioni per mostrare come vengono stampate.

```
Soluzione 1:
+-+-+-+-+-+-+-+-+
|Q| | | | | | | |
+-+-+-+-+-+-+-+-+
| | | | |Q| | | |
+-+-+-+-+-+-+-+-+
| | | | | | | |Q|
+-+-+-+-+-+-+-+-+
| | | | | |Q| | |
+-+-+-+-+-+-+-+-+
| | |Q| | | | | |
+-+-+-+-+-+-+-+-+
| | | | | | |Q| |
+-+-+-+-+-+-+-+-+
| |Q| | | | | | |
+-+-+-+-+-+-+-+-+
| | | |Q| | | | |
+-+-+-+-+-+-+-+-+
```