

Andrea Augello

Department of Engineering, University of Palermo, Italy

Simulazione quesiti di esame



Simulazione quesiti di esame

In questa esercitazione vedremo due quesiti sulla ricerca che sono stati proposti in esami passati.

Quesito 1

Quesito 1

Si implementi in python un programma che risolva il seguente problema di ricerca: Sia data una scacchiera $1 \times N$ contenente delle pedine con un lato nero e uno bianco tali che ci siano i pedine con il lato nero rivolto verso l'alto e j pedine con il lato bianco rivolto verso l'alto, con $i + j < N$.

Trovare **la più breve** sequenza di mosse (spostamenti) che permetta di rendere tutte le pedine nere applicando le seguenti regole:

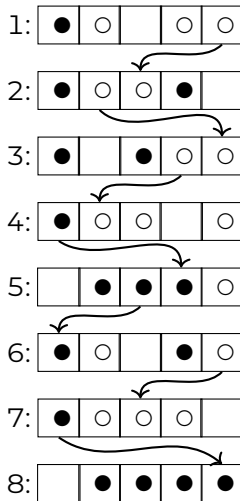
- ▶ Le pedine possono essere spostate solo su spazi vuoti saltando sopra sequenze di pedine adiacenti.
- ▶ È possibile saltare oltre più pedine adiacenti.
- ▶ Non è possibile saltare oltre spazi vuoti.
- ▶ Quando si salta sopra delle pedine, queste vengono ribaltate (cambiando il loro colore da nero a bianco o viceversa).
- ▶ La pedina che viene spostata rimane dello stesso colore.

Esempio

Prendiamo in considerazione una configurazione con una scacchiera 1×5 , con le pedine disposte come segue:



La sequenza di mosse più breve consiste nel muovere le pedine in posizione 5, 2, 4, 1, 3, 5, 1 ottenendo i seguenti stati intermedi:



Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è una configurazione della scacchiera, ovvero una sequenza di N elementi ciascuno dei quali può avere uno dei tre valori: bianco, nero o vuoto.

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è una configurazione della scacchiera, ovvero una sequenza di N elementi ciascuno dei quali può avere uno dei tre valori: bianco, nero o vuoto.
- ▶ **Qual è lo stato iniziale?** È la configurazione iniziale della scacchiera, data come input.

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è una configurazione della scacchiera, ovvero una sequenza di N elementi ciascuno dei quali può avere uno dei tre valori: bianco, nero o vuoto.
- ▶ **Qual è lo stato iniziale?** È la configurazione iniziale della scacchiera, data come input.
- ▶ **Qual è lo stato obiettivo?** Non lo conosciamo in maniera esplicita, ma sappiamo che caratteristiche deve avere: tutte le pedine devono essere nere (ovvero non devono esserci elementi con valore 1).

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è una configurazione della scacchiera, ovvero una sequenza di N elementi ciascuno dei quali può avere uno dei tre valori: bianco, nero o vuoto.
- ▶ **Qual è lo stato iniziale?** È la configurazione iniziale della scacchiera, data come input.
- ▶ **Qual è lo stato obiettivo?** Non lo conosciamo in maniera esplicita, ma sappiamo che caratteristiche deve avere: tutte le pedine devono essere nere (ovvero non devono esserci elementi con valore 1).
- ▶ **Quali sono le azioni possibili?** Per ogni pedina, possiamo spostarla nello spazio vuoto più vicino a destra o a sinistra (se non è adiacente), invertendo il colore delle pedine nelle celle attraversate(saltate).

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è una configurazione della scacchiera, ovvero una sequenza di N elementi ciascuno dei quali può avere uno dei tre valori: bianco, nero o vuoto.
- ▶ **Qual è lo stato iniziale?** È la configurazione iniziale della scacchiera, data come input.
- ▶ **Qual è lo stato obiettivo?** Non lo conosciamo in maniera esplicita, ma sappiamo che caratteristiche deve avere: tutte le pedine devono essere nere (ovvero non devono esserci elementi con valore 1).
- ▶ **Quali sono le azioni possibili?** Per ogni pedina, possiamo spostarla nello spazio vuoto più vicino a destra o a sinistra (se non è adiacente), invertendo il colore delle pedine nelle celle attraversate(saltate).
- ▶ **Quali informazioni ci serviranno durante la ricerca?** Lo stato corrente e la storia degli stati visitati.

Suggerimenti

- ▶ Un modo efficace per rappresentare lo stato è usare una lista di interi, per esempio utilizzando -1 per le pedine nere, 1 per le pedine bianche e 0 per gli spazi vuoti.
- ▶ Le azioni possono essere rappresentate come una tupla di indici i cui contenuti vanno scambiati.
- ▶ Una mossa è valida solo se i due indici non sono consecutivi, uno dei due corrisponde al valore 0, e se, estremi esclusi, tutti gli elementi tra i due indici sono diversi da 0.
- ▶ Nella transizione di stato, conviene gestire separatamente lo scambio degli elementi e l'inversione del colore delle pedine saltate.

Quesito 2

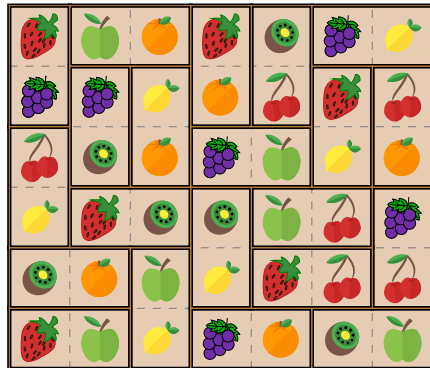
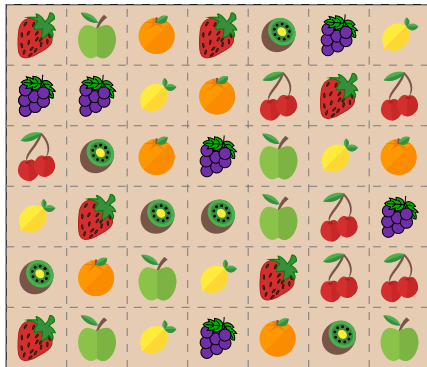
Quesito 2

Si implementi in python un programma che risolva il seguente problema di ricerca: Una crostata è stata condita con sette diversi tipi di frutta distribuiti su una griglia.

Si vuole suddividere la crostata suddividendola in pezzi contigui di dimensione da due celle facendo in modo che non ci siano due pezzi di crostata con la stessa combinazione di frutti. Il programma deve essere in grado di trovare **tutte** le possibili soluzioni valide.

Esempio

Data la crostata mostrata in figura, una possibile soluzione è quella evidenziata con le linee spesse.



Nota bene: A seconda della configurazione della crostata (ovvero della disposizione dei frutti e della dimensione della crostata), potrebbero esserci più soluzioni valide o nessuna.

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è una configurazione parziale della crostata, ovvero una suddivisione parziale della crostata in pezzi di due celle.

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è una configurazione parziale della crostata, ovvero una suddivisione parziale della crostata in pezzi di due celle.
- ▶ **Qual è lo stato iniziale?** La crostata intera, senza alcuna suddivisione.

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è una configurazione parziale della crostata, ovvero una suddivisione parziale della crostata in pezzi di due celle.
- ▶ **Qual è lo stato iniziale?** La crostata intera, senza alcuna suddivisione.
- ▶ **Qual è lo stato obiettivo?** Non lo conosciamo in maniera esplicita, ma sappiamo che caratteristiche deve avere: la crostata deve essere completamente suddivisa in pezzi di due celle, e non ci devono essere due pezzi con la stessa combinazione di frutti.

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è una configurazione parziale della crostata, ovvero una suddivisione parziale della crostata in pezzi di due celle.
- ▶ **Qual è lo stato iniziale?** La crostata intera, senza alcuna suddivisione.
- ▶ **Qual è lo stato obiettivo?** Non lo conosciamo in maniera esplicita, ma sappiamo che caratteristiche deve avere: la crostata deve essere completamente suddivisa in pezzi di due celle, e non ci devono essere due pezzi con la stessa combinazione di frutti.
- ▶ **Quali sono le azioni possibili?** Scegliere due celle adiacenti (in orizzontale o in verticale) che non siano già state assegnate a un pezzo di crostata, e formare un nuovo pezzo di crostata con esse, a patto che la combinazione di frutti non sia già stata usata in precedenza.

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è una configurazione parziale della crostata, ovvero una suddivisione parziale della crostata in pezzi di due celle.
- ▶ **Qual è lo stato iniziale?** La crostata intera, senza alcuna suddivisione.
- ▶ **Qual è lo stato obiettivo?** Non lo conosciamo in maniera esplicita, ma sappiamo che caratteristiche deve avere: la crostata deve essere completamente suddivisa in pezzi di due celle, e non ci devono essere due pezzi con la stessa combinazione di frutti.
- ▶ **Quali sono le azioni possibili?** Scegliere due celle adiacenti (in orizzontale o in verticale) che non siano già state assegnate a un pezzo di crostata, e formare un nuovo pezzo di crostata con esse, a patto che la combinazione di frutti non sia già stata usata in precedenza.
- ▶ **Quali informazioni ci serviranno durante la ricerca?** Dovremo tenere traccia di quali fette di crostata sono state assegnate a quali pezzi (varia durante l'esecuzione), e che frutto si trova in ogni cella (costante).

Suggerimenti

- ▶ La griglia iniziale può essere una lista di liste (associando un numero ad ogni frutto) mantenuta come attributo dell'agente.
- ▶ Lo stato può essere una seconda matrice (inizialmente tutta a zero) che indica per ogni cella a quale pezzo di crostata appartiene (0 se non è ancora stato assegnato a nessun pezzo).
- ▶ Per la verifica dell'correttezza di uno stato, i set possono essere utili.
- ▶ Si consiglia di affettare la crostata iniziando da uno degli angoli, e procedendo in modo sistematico (ad esempio, da sinistra a destra e dall'alto al basso).

Quesito 3 (bonus)

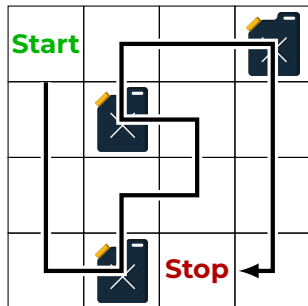
Quesito 3 (bonus)

Sia dato un mondo a griglia $N \times N$ in cui un agente si può muovere in quattro direzioni (su, giù, sinistra, destra). L'agente ha un serbatoio di carburante che consente un'autonomia di spostamento pari al più a p passi. Alcune celle della griglia, note a priori, contengono del carburante che l'agente può raccogliere per riportare il proprio serbatoio al massimo (il carburante in eccesso viene sprecato).

Si realizzi un programma in Python che consenta ad un agente di determinare (se esiste) un percorso che, da una cella di partenza, gli permetta di percorrere l'intera griglia, attraversando ogni cella esattamente una volta, senza mai esaurire il carburante.

Esempio

Data una griglia di dimensione 4×4 , con il carburante collocato nelle celle (1, 4), (2, 2) e (4, 2), un agente con $p = 4$ che inizia dall'angolo in alto a sinistra (cella 1, 1) con il serbatoio pieno, può seguire il percorso mostrato nella figura seguente, visitando tutte le celle della griglia fermandosi nella cella stop (cella 4, 3) senza esaurire il carburante.



Nota bene: La cella di stop può essere una qualsiasi cella della griglia. L'importante è che l'agente riesca a visitare tutte le celle della griglia.

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è dato dalla posizione corrente dell'agente e dalle celle già visitate (il livello di carburante può fare parte dello stato o può essere calcolato di volta in volta).

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è dato dalla posizione corrente dell'agente e dalle celle già visitate (il livello di carburante può fare parte dello stato o può essere calcolato di volta in volta).
- ▶ **Qual è lo stato iniziale?** La posizione iniziale dell'agente (data come input) e nessuna cella visitata.

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è dato dalla posizione corrente dell'agente e dalle celle già visitate (il livello di carburante può fare parte dello stato o può essere calcolato di volta in volta).
- ▶ **Qual è lo stato iniziale?** La posizione iniziale dell'agente (data come input) e nessuna cella visitata.
- ▶ **Qual è lo stato obiettivo?** Non lo conosciamo in maniera esplicita, ma sappiamo che caratteristiche deve avere: tutte le celle della griglia devono essere state visitate esattamente una volta.

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è dato dalla posizione corrente dell'agente e dalle celle già visitate (il livello di carburante può fare parte dello stato o può essere calcolato di volta in volta).
- ▶ **Qual è lo stato iniziale?** La posizione iniziale dell'agente (data come input) e nessuna cella visitata.
- ▶ **Qual è lo stato obiettivo?** Non lo conosciamo in maniera esplicita, ma sappiamo che caratteristiche deve avere: tutte le celle della griglia devono essere state visitate esattamente una volta.
- ▶ **Quali sono le azioni possibili?** Muoversi in una delle quattro direzioni (se non si esce dalla griglia, se la cella non è già stata visitata, e se si ha carburante sufficiente per farlo).

Poniamoci le solite domande

- ▶ **Qual è lo spazio di stati?** Ogni stato è dato dalla posizione corrente dell'agente e dalle celle già visitate (il livello di carburante può fare parte dello stato o può essere calcolato di volta in volta).
- ▶ **Qual è lo stato iniziale?** La posizione iniziale dell'agente (data come input) e nessuna cella visitata.
- ▶ **Qual è lo stato obiettivo?** Non lo conosciamo in maniera esplicita, ma sappiamo che caratteristiche deve avere: tutte le celle della griglia devono essere state visitate esattamente una volta.
- ▶ **Quali sono le azioni possibili?** Muoversi in una delle quattro direzioni (se non si esce dalla griglia, se la cella non è già stata visitata, e se si ha carburante sufficiente per farlo).
- ▶ **Quali informazioni ci serviranno durante la ricerca?** La posizione corrente dell'agente, le celle già visitate, e la posizione delle celle con carburante (quest'ultima può essere mantenuta come attributo dell'agente).