Esercitazioni di Informatica B

Ricorsione

Stefano Cereda
stefano.cereda@polimi.it

11/12/2018

Politecnico di Milano

Competizione Forza4 i

Vi si richiede di implementare una strategia in MATLAB per giocare automaticamente a Forza4.

- La strategia deve essere realizzata interamente con una funzione MATLAB, senza interazione con l'utente
- La funzione che implementa la strategia deve avere come signature $col = strat_matricola(Griglia, colore)$ deve prendere in ingresso una matrice *Griglia* (di dimensioni 6x7), che rappresenta la griglia di gioco, ed un intero colore, di valore 1 o -1, che rappresenta il colore della pedina da giocare (rosso o giallo) e restituire un intero *col*, con valore tra 1 e 7, che indica il numero della colonna in

cui inserire una pedina. Perché la mossa sia valida, *col* deve indicare una colonna in cui c'è ancora spazio (ci sono meno di sei pedine).

Una griglia rappresentante la situazione di una partita di Forza4 è modellizzata da una matrice 6x7 contenente solo i valori -1, 0, 1. 0 rappresenta una casella vuota, 1 una pedina del primo giocatore (rosso) e -1 una del secondo giocatore (giallo).

Forza4 - Esempio strategia randomica

```
1 function col = random_strategy(Griglia, colore
      )
2 % La strategia mette la pedina in una colonna
      libera a caso
3 % (e quindi non utilizza il colore)
4
5 % trovo le colonne con ancora posto
6 cols = find(Griglia(1,:) == 0);
7
8 % ne seleziono una a caso
9 col = cols(randi(length(cols)));
```

Forza4 - Test strategia

Testiamo la strategia randomica tramite il file main.m disponibile su beep.

La funzione di cui è richiesta l'implementazione verrà richiamata nello script nel file main.m. Lo script vi chiederà di inserire il nome della vostra funzione (che deve essere visibile nel path di MATLAB) e il nome della strategia contro cui volete giocare. Ci sono due opzioni già disponibili nel codice messo a disposizione: una strategia che richiede l'intervento dell'utente (human_strategy()) e una che seleziona una mossa valida secondo un criterio random (random_strategy())

Forza4 - Regole i

Le regole della competizione si trovano su beep, corso Competitions Informatica B.

- È necessario usare operazioni vettoriali, si possono usare cicli, si può ricorrere a randomizzazione e ricorsione
- Una giocata non può richiedere più di 3 secondi su un processore di un laptop comune (per evitare possibili loop infiniti). In caso contrario, la partita viene vinta a tavolino dall'avversario

Forza4 - Regole ii

- Una giocata deve essere valida, cioè deve corrispondere all'inserimento di una pedina in una colonna della griglia in cui è ancora disponibile spazio. In caso contrario, la partita viene vinta a tavolino dall'avversario
- La funzione non deve restituire errori. In caso contrario, la partita viene vinta a tavolino dall'avversario
- Verrà valutata la strategia grazie ad un torneo all'italiana tra tutti i partecipanti (di tutti gli scaglioni)
- Potete presentare una sola strategia a persona

Forza4 - Regole iii

- Entro la scadenza indicata (19/12/2018, ore 23.59), caricate il file .m contenente la vostra strategia (opportunamente commentata) nell'apposita cartella di consegna. Il nome della funzione deve essere compatibile con le regole indicate qui sopra. Nel commento in testa al file specificate il vostro nome e cognome e il vostro docente di Informatica B.

Notazione scientifica

Scrivere una funzione iterativa che dato un numero reale positivo lo traduca in notazione scientifica.

$123.456 = 1.23456E2$

Scrivere poi la stessa funzione in modo ricorsivo.

Triangolo di Tartaglia

Si consideri una matrice triangolare T rappresentante il triangolo di Tartaglia, i cui valori sono definiti nel seguente modo:

- T(r,1) = 1
- T(r,r) = 1
- T(r,c) = T(r-1, c-1) + T(r-1, c)

Si scriva una funzione ricorsiva che, usando la definizione sopra, permetta di calcolare il valore di un generico elemento della matrice T a partire dagli indici di riga e colonna.

Funzione misteriosa 1

Si dica cosa calcola la seguente funzione ricorsiva quando si passano i parametri 7 e 2 ed in generale.

```
1  function [z]=mistero1(x,y)
2      if x<y
3          z=0;
4      else
5          if mod(x,y) == 0
6              z=x+mistero1(x-y,y);
7          else
8              z=mistero1(x-1,y);
9          end
10     end
```


La funzione calcola la somma di tutti i multipli di y compresi

Funzione misteriosa 2

Si dica cosa calcola la seguente funzione ricorsiva quando si passa il parametro 7, quando si passa il parametro 8 ed in generale.

```
1 function [z] = mistero2(x)
2     if x>= 1
3         z=mod(x,2) + 10*mistero2(floor(x/2))
4     else
5         z=0;
6     end
```

mistero2(x) calcola un valore numerico in base 10 la cui sequenza di cifre può essere interpretata come rappresentazione del numero naturale x nel sistema binario.

Funzione misteriosa 3 - TdE 17/02/2009 i

Si consideri il seguente programma:

```
1  function[ris] = s(n)
2      if n<1
3          ris = -1;
4      elseif n>=1 & n<=4 %*
5          ris = n; %*
6      else
7          ris = s(n-2)*s(n-4);
8      end
9  %script di chiamata
10 for x = 1:8
11     fprintf('%d, ', s(x));
12 end
```

Funzione misteriosa 3 - TdE 17/02/2009 ii

Quali risultati vengono stampati a video?

Si supponga ora di rimuovere le linee evidenziate da un asterisco:

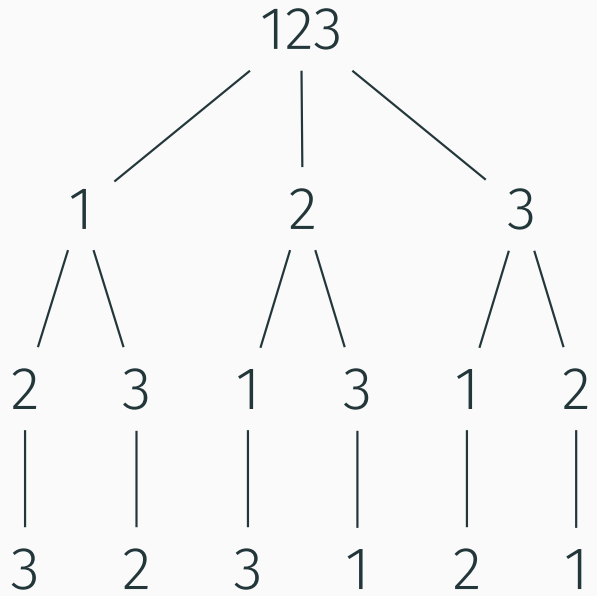
1. La terminazione della ricorsione è ancora garantita per ogni valore dell'argomento n? In caso affermativo giustificare brevemente la risposta, in caso negativo riportare almeno un esempio di argomento in cui la ricorsione non termina.
2. Con la modifica apportata quali risultati vengono stampati a video?

Permutazioni i

Si scriva una funzione in MATLAB che abbia come parametro di ingresso un vettore contenente un insieme di valori, e che stampi tutte le permutazioni di tali valori.

Le permutazioni di [1 2 3] sono: [1 2 3], [1 3 2], [2 1 3], [2 3 1], [3 1 2], [3 2 1].

Permutazioni ii



Permutazioni iii

Osservando lo schema possiamo notare che abbiamo bisogno di due informazioni: i valori già considerati precedentemente nelle permutazioni (per non ripeterli) e i valori ancora disponibili (da aggiungere). Queste informazioni dovranno quindi essere i parametri d'ingresso della funzione ricorsiva. Tuttavia, la consegna dice di scrivere una funzione che riceva come parametro solamente il vettore di valori da permutare, dovremo quindi aggiungere una funzione per mascherare il cambio di parametri

Permutazioni iv

La funzione si struttura nel seguente modo:

- Caso base L'insieme dei valori da inserire è vuoto: stampo i valori già inseriti e termino.
- Ricorsione L'insieme dei valori da inserire non è vuoto: scandisco tutti i possibili valori ancora da inserire ed eseguo una invocazione ricorsiva della funzione per ognuno di questi valori. In queste invocazioni ricorsive passo come primo parametro i valori già inseriti dal chiamante, più quello nuovo che sto scandendo. Come secondo parametro invece passo tutti i valori ancora da inserire, eccetto quello che sto scandendo.

Vettore triplo - Prova in itinere 2014 i

Un vettore *x* si definisce vettore triplo di un altro vettore *y* quando *x* e *y* sono di uguale lunghezza e ogni elemento di *x* è maggiore del triplo del corrispondente elemento di *y*.

Scrivere in Matlab/Octave una funzione ricorsiva *vettoreTriplo* che riceve in ingresso due vettori colonna *a* e *b*. La funzione vettoreTriplo:

1. Restituisce -1 se *a* e *b* hanno lunghezza diversa, o se non sono due vettori colonna.
2. Restituisce 0 se *a* e *b* sono due vettori colonna di uguale lunghezza, ma non è vero che ogni elemento di *a* è maggiore del triplo del corrispondente elemento di *b*.

Vettore triplo - Prova in itinere 2014 ii

3. Restituisce 1 se *a* e *b* sono due vettori colonna di uguale lunghezza e ogni elemento di *a* è maggiore del triplo del corrispondente elemento di *b*.

Esempi:

- vettoreTriplo ([1; 2; 3; 4], [2 5 1 10]) restituisce -1
- vettoreTriplo ([1 2 3 4; 5 3 5 5], [2; 2; 3; 9]) restituisce -1
- vettoreTriplo ([7; 2; 4; 4], [2; 5; 1; 10]) restituisce 0
- vettoreTriplo([7; 16; 40; 31], [2; 5; 13; 10]) restituisce 1

Vettore triplo - Prova in itinere 2014 iii

Nell'implementazione non è consentito utilizzare cicli, e quindi non si possono usare le istruzioni *while* e *for*. Inoltre, non è possibile utilizzare alcuna funzione di libreria disponibile in Matlab/Octave ad eccezione di *length* e *size*.

Funzione misteriosa 4 - TdE 17/09/2012 i

Si consideri la seguente funzione ricorsiva in linguaggio MATLAB:

```
function n = calcola(x)
if (floor(x/10) == 0)
    n = 1;
else
    n = 1 + calcola(floor(x/10));
end
```

Rispondere ai seguenti quesiti.

Funzione misteriosa 4 - TdE 17/09/2012 ii

1. Simulare l'esecuzione della funzione e indicare il valore restituito in corrispondenza di ciascuna delle chiamate: calcola(4) e calcola(123)
2. Assumendo che la funzione calcola riceva come parametro un qualsiasi numero intero positivo n , dire qual è il risultato dell'esecuzione di calcola(n) al variare di n. Motivare adeguatamente la risposta.