

INFORMATICA GENERALE

Homework 2: 2048

docente: IVANO SALVO
Sapienza Università di Roma

pubblicazione: 7.V.2014 - consegna 19.V.2014

Probabilmente conoscete tutti il gioco del 2048¹. Una matrice 4×4 contiene numeri che sono tutti potenze di due o caselle vuote. Si può muovere a destra, sinistra, alto e basso, causando lo “scivolamento” di tutti i numeri della matrice, facendoli occupare la posizione libera più a destra, sinistra, alto o basso libera. Ad esempio, muovendo verso destra, ogni numero occuperà la casella più a destra possibile: questo dipende da quante caselle vuote ci sono a destra, e da eventuali “fusioni”: infatti, se due numeri uguali si trovano adiacenti (sulla stessa riga) muovendo verso destra (o sinistra) si fondono e la casella conterrà poi la somma dei due numeri. Muovendo verso l’alto o il basso vengono fusi numeri adiacenti sulla stessa colonna. Alla fine di ogni mossa compare nondeterministicamente un 2 o un 4 in qualche casella libera. L’obiettivo del gioco è, a forza di fusioni, di ottenere il numero 2048 (o anche di più). Ogni volta che due numeri si fondono in unico numero (ad esempio due 4 si fondono per generare un 8) si ottengono un numero di punti pari al nuovo numero generato (nel nostro esempio, 8).

Noi codificheremo in una matrice di interi 4×4 una posizione del 2048, con la convenzione che le caselle vuote siano rappresentate dal numero 0.

Esercizio 1 Il gioco finisce quando non si possono fare ulteriori mosse: ciò accade quando si verificano entrambe le due seguenti condizioni:

1. non ci sono caselle libere;
2. non c’è nessuna possibile fusione, in nessuna direzione (cioè non ci sono coppie di numeri uguali adiacenti in orizzontale o in verticale).

¹vedi ad esempio [http://it.wikipedia.org/wiki/2048_\(videogioco\)](http://it.wikipedia.org/wiki/2048_(videogioco))

Scrivere un programma C che, letta da input una matrice 4×4 di interi, restituisca 1 se il gioco è finito e 0 altrimenti.

Esercizio 2 Supponendo che tutti i numeri apparsi durante il gioco siano inizialmente dei 2, il punteggio ottenuto potrebbe essere ricostruito semplicemente analizzando i numeri che compaiono nella matrice². Infatti, un numero 2^n ($n > 1$) è stato ottenuto fondendo prima 2^{n-1} 2, poi i 2^{n-2} 4 e così via. È facile convincersi che i punti ottenuti per ottenere il numero 2^n sono $p(2^n)$, dove la funzione p è definita induttivamente sulle potenze di 2 come segue:

$$\begin{aligned} p(2) &= 0 \\ p(2^n) &= 2p(2^{n-1}) + 2^n \quad (\text{per } n > 1) \end{aligned}$$

Scrivere un programma C che, letta da input una matrice 4×4 di interi, restituisce il punteggio ottenuto a quel punto (sarà $\sum_{i=0}^3 \sum_{j=0}^3 p(a_{ij})$).

Esercizio 3 (FACOLTATIVO – vale un bonus di 1 punto sul voto finale)

Scrivere un programma C che, letta da input una matrice 4×4 di interi, e un numero d intero tra 0 e 3, interpreta la matrice come una posizione del 2048, d come la direzione della mossa (0=destra, 1=sinistra, 2=alto e 3=basso) e stampa la matrice dopo l'esecuzione della mossa (senza far apparire nessun nuovo numero come invece accade nel gioco) e il punteggio ottenuto.

Ad esempio, leggendo la matrice:

```
2  4  2  0
2  4  8  0
4 16 32  0
8 16 32 128
```

e la direzione 3 (basso), restituisca il seguente output:

```
0  0  0  0
4  0  2  0
4  8  8  0
8 32 64 128
```

108

infatti, si sono ottenuti per fusione un 64, un 32, un 8 e un 4 e $64+32+8+4=108$.

Onde evitare errori, usate le funzioni di input/output contenute nelle note.

²ciò non è del tutto vero nel gioco reale, in quanto a volte possono apparire dei 4.