Esercitazione 3

Cammino nel labirinto (“Algoritmi e programmazione in pratica: da specifiche a codice C”, sez. 1.3.4)

Problema di verifica:

- Data una soluzione (sequenza di mosse), verificare che corrisponda ad un cammino valido
- Verificare che il cammino sia semplice
- Calcolare la lunghezza del cammino

Esercitazione 3

Il labirinto e' memorizzato in un file sotto forma di matrice di caratteri

Il numero di righe / colonne sono specificate nel file (massimo 50)

La matrice di caratteri codifica caselle libere ('-') e muri ('X')

La casella di ingresso si trova in alto a sinistra ('I'), quella di uscita in basso a destra ('U')

Esercitazione 3

Le mosse sono memorizzate in un secondo file

Le mosse sono rappresentate da spostamenti di X caselle in una direzione, diagonali incluse (es. $+3 \ -3$)

Dobbiamo verificare che gli spostamenti siano validi

Dobbiamo verificare che il cammino sia semplice (ogni casella è visitata al più una volta)

Dobbiamo verificare che il punto di arrivo sia l'uscita

Dobbiamo misurare la lunghezza del cammino (numero di passi)

Esercitazione 3

Due soluzioni

In entrambi i casi:

- Cammino non acquisito in modo esplicito
- La funzione di acquisizione si occupa anche delle verifiche
- La verifica di cammino semplice viene realizzata marcando le caselle in $O(L)$ (invece di memorizzare il cammino e successivamente verificare che non contenga punti ripetuti, $O(L^2)$)

Esercitazione 3

Prima soluzione: la verifica di mossa valida viene fatta in modo esplicito per tutte le direzioni

Seconda soluzione: si parametrizzano le direzioni, rappresentando gli spostamenti come N passi di lunghezza 1 lungo una direzione

Operazioni su insiemi

Operazioni su insiemi (“Algoritmi e programmazione in pratica: da specifiche a codice C”, sez. 1.1.4)

- Insiemi rappresentati da vettori
- Gestione elementi ripetuti
- Vogliamo effettuare operazioni insiemistiche (unione, intersezione, differenza simmetrica)

Operazioni su insiemi

Soluzione 1:

- Vettori non ordinati
- Utilizzo di una funzione di *appartenenza*
- Complessità quadratica

Soluzione 2:

- Vettori ordinati
- Le operazioni sfruttano l'ordinamento per fornire la soluzione in tempo lineare
- Complessità dipendente dall'algoritmo di ordinamento scelto