



Commento al Laboratorio n. 7

Esercizio n. 1: Gioco di ruolo (multi-file, con ADT)

La soluzione all'esercizio segue direttamente le strategie proposte per la realizzazione degli ADT, tenendo presente le linee-guida date nelle specifiche.

Esercizio n.2: Corpo libero

Strutture dati: si definiscono le seguenti strutture dati:

- per i dati:
 - un tipo `elemento_t` come `struct` che contiene i campi dell'elemento
 - un tipo `diagonale_t` come `struct` che contiene i campi della diagonale, incluso un vettore `elementi` di indici che permettono di accedere alla collezione di elementi
- per le collezioni di dati:
 - un tipo `elementi_t` come `struct` che contiene un vettore di dati di tipo `elemento_t` e il numero di elementi disponibili
 - un tipo `diagonali_t` come `struct` che contiene un vettore di dati di tipo `diagonale_t`, il numero `nd` di diagonali valide effettivamente disponibili e la dimensione massima `maxd` del vettore allocato, per tracciare la crescita dinamica del vettore durante la fase di generazione di tutte le diagonali ammissibili.

Analisi del problema: la soluzione si articola in 3 fasi:

1. inizializzazione della collezione `D` delle diagonali, lettura degli elementi e loro memorizzazione nella collezione di dati `E` degli elementi
2. generazione delle diagonali a partire dagli elementi
3. generazione del programma ottimo a partire dalle diagonali disponibili.

Modelli:

- generazione delle diagonali a partire dagli elementi: si osservi che non è vietato esplicitamente ripetere lo stesso elemento in una diagonale. Si osservi inoltre che l'ordine con gli elementi compaiono nella diagonale conta, in quanto le specifiche impongono vincoli (ad esempio di entrata/uscita, precedenza, sequenza finale etc.). Il modello è quindi quello delle disposizioni ripetute di `MAXE` elementi presi al massimo a 5 a 5. Per limitare lo spazio di ricerca si impongono condizioni sulla discesa ricorsiva (pruning) per escludere di superare la massima difficoltà ammessa per una diagonale, per forzare la condizione di ingresso e quella di precedenza per il primo elemento della diagonale. La condizione di terminazione verifica sia se sono stati presi esattamente 5 elementi, ma anche se si può interrompere la costruzione della diagonale anticipatamente (con meno di 5 elementi) in quanto l'ultimo non prevede elementi che lo seguano e poiché, purché valida per composizione, anche diagonali più corte di 5 elementi sono comunque ammissibili. La soluzione corrente (`diag`, vettore di indici che permettono di accedere agli elementi che appartengono alla soluzione) viene memorizzata nella collezione di dati `D` delle diagonali posto che sia soddisfatto il vincolo sul numero di elementi acrobatici. Il vettore `vd` in `D` viene riallocato dinamicamente in funzione del numero di diagonali generate
- generazione del programma ottimo a partire dalle diagonali disponibili: si osservi che anche in questo caso non è vietato esplicitamente ripetere la stessa diagonale. La specifica secondo cui,



**Politecnico
di Torino**

03AAX ALGORITMI E STRUTTURE DATI CORSO DI LAUREA IN INGEGNERIA INFORMATICA A.A. 2023/24

se il ginnasta include un elemento finale di difficoltà 8 o superiore nell'ultima diagonale presentata in gara, il punteggio complessivo della diagonale viene moltiplicato per 1.5 sembra orientare verso un modello in cui conta l'ordine. In realtà basta verificare che vi sia una diagonale con elemento finale di difficoltà 8 o superiore e considerarla come ultima per usufruire del bonus moltiplicativo. Pertanto, a differenza del punto precedente, non conta l'ordine con cui si inseriscono le diagonali nel programma. Il modello sottostante è quindi quello delle combinazioni ripetute. L'output però non riporta le diagonali nell'ordine voluto. Il pruning dello spazio è realizzato subordinando la discesa ricorsiva alla verifica che non si ecceda la massima difficoltà del programma. La validità della soluzione è verificata nella condizione di terminazione.