

Relazione: laboratorio n° 10, esercizio n° 2

Struttura dati scelta: il programma è stato scomposto in due file modulari; il file “main.c” contiene la funzione principale e sostanzialmente coordina le operazioni, mentre il file “ott_combsem.c”, interfacciato dall’header “ott_combsem.h”, è quello che permette effettivamente di risolvere il problema. Dal momento che il client non ha bisogno di manipolare direttamente i dati del problema – al di fuori delle dimensioni dell’inventario – questo secondo file è stato realizzato come quasi-ADT. I nuovi tipi di dato – una struct “toggetto” per rappresentare il generico oggetto della lista complessiva e una “t_scelta” per rappresentare il generico oggetto inserito in soluzione – sono quindi definiti interamente nel file .c, così come sono definite nel file .c nella forma di variabili globali di tipo static le variabili necessarie per contenere i vari dati, tra cui vanno notati il vettore dinamico di togetto “vScelte”, che conterrà la lista completa degli oggetti disponibili, i vettori dinamici di t_scelta “vSol” e “vSolCorr”, che conterranno rispettivamente la soluzione ottima e la soluzione temporanea, e la matrice dinamica di interi “inv”, che sarà utilizzata per simulare la sistemazione degli oggetti scelti nell’inventario.

N.B.: la struct togetto contiene le informazioni fornite su ciascun oggetto dal file “equip.txt”, a cui vengono aggiunte informazioni sull’area dell’oggetto; la struct t_scelta contiene l’indice dell’oggetto nel vettore vScelte più le informazioni sulla sistemazione dell’oggetto nell’inventario sotto forma di coordinate del suo angolo più in alto a sinistra.

Scelte algoritmiche: siccome ogni oggetto può essere inserito in soluzione una volta sola e siccome la soluzione non è posizionale – ovvero l’ordine degli oggetti in soluzione non conta – si è deciso di modellizzare la costruzione delle soluzioni mediante le combinazioni semplici, aggiungendo del parziale pruning: in particolare un oggetto viene inserito nella potenziale soluzione – sotto forma di indice dell’oggetto nel vettore vScelte – solo se le sue dimensioni sono compatibili con quelle dell’inventario (un oggetto che occupa 7 righe non può essere inserito in un inventario che di righe ne prevede 6) e se l’area che occupa è minore o uguale a quella attualmente ancora libera nell’inventario. Quando tutti gli oggetti sono stati esaminati, la soluzione potenziale corrente è considerata completa e se ne valuta l’utilità: se essa è maggiore dell’utilità utTot dell’attuale soluzione ottima (notare che essendo static utTot è automaticamente inizializzata a 0), si procede ulteriormente e si controlla, mediante la funzione oggiRichiesti, che nella soluzione siano contemplate almeno un’arma, un’armatura e una pozione. Se anche questo passo dà risposta affermativa si passa all’ultimo controllo, ovvero l’inserimento degli oggetti nell’inventario (rappresentato dalla matrice inv) attraverso la funzione disponiOgg. Se gli oggetti vengono disposti nella matrice con successo, la soluzione corrente diventa la nuova soluzione ottimale e viene quindi registrata nel vettore vSol; allo stesso modo viene aggiornato utTot, in cui viene inserita l’utilità complessiva della nuova soluzione ottimale. Ciò va avanti fino a quando non sono state provate tutte le possibili combinazioni di oggetti; al termine del programma viene poi fornito l’esito della ricerca.

N.B.: per la sistemazione degli oggetti nella matrice il programma si appoggia a un vettore nel quale la soluzione corrente viene registrata ordinando gli oggetti che ne fanno parte per area decrescente. A questo punto gli oggetti così ordinati vengono inseriti uno alla volta nella matrice, partendo dalla casella della matrice più in alto a sinistra e spostandosi da sinistra a destra e dall’alto in basso. In questo modo si ha garanzia di non sprecare spazio, poiché gli oggetti che ne richiedono di più vengono disposti per primi. Notare che con “inserimento” di un oggetto nella matrice si intende la scrittura nelle caselle della matrice occupate dall’oggetto dell’indice dell’oggetto stesso nel vettore vScelte.