

ESERCIZIO TORRE DI CARTONI

All'interno di un magazzino ci sono n pacchi. Ogni pacco è caratterizzato da un peso, un'altezza, e da un limite massimo di peso che il pacco può sostenere sopra di sé. Pesi, altezze e limiti sono interi positivi.

Per rappresentare queste caratteristiche si utilizzi la struttura:

```
typedef struct {
    unsigned peso;
    unsigned altezza;
    unsigned limite;
} pacco;
```

Scrivere una funzione:

```
void CostruisciTorre(int n, pacco *pacchi, int s, torre *corrente, torre *best, int *listaUso);
```

che, dato un array di n pacchi, calcoli qual è l'altezza massima di una pila di pacchi che può essere formata con essi rispettando il vincolo che nessun pacco abbia sopra di sé un peso superiore al limite consentito. I parametri della funzione `CostruisciTorre` sono elencati di seguito:

`n`: numero di pacchi nell'array,

`pacchi`: array di pacchi disponibili,

`s`: posizione corrente (a che livello dell'albero di backtrack si trova la funzione corrente),

`corrente`: torre corrente,

`best`: miglior torre,

`listaUso`: vettore binario (0-1) che tiene traccia dei pacchi utilizzati (=1) e non (=0) del vettore dei pacchi.

Il tipo `torre` è definito come segue:

```
typedef struct {
    unsigned altezza;
    int *pacchiTorre;
} torre;
```

dove `altezza` rappresenta l'altezza della torre e `pacchiTorre` è un puntatore ad un array che contiene la lista dei pacchi utilizzati per costruire la torre (lista delle posizioni). Ad esempio, se nell'array dei `pacchiTorre` di una torre troviamo i valori 0, 2, 1 significa che la torre è costruita con i pacchi in posizione 0, 2 e 1 dell'array di pacchi. Suggerimento: si utilizzi il valore -1 per modellare il caso "nessun pacco".