

Esercizio 1 di martedì 5/5/2020

Il problema generalizza ancora l'esercizio del campo minato. Ora abbiamo un array $\text{int } X[n*n]$ di interi maggiori o uguali a 0 (non più di bool) che vedremo come un array $\text{int}[n][n]$ dove lo 0 rappresenta ancora le mine e i valori >0 indicano il valore della casella safe in cui sono contenuti. Diremo che un cammino in questo campo minato che attraversa solo caselle safe ha come valore la somma dei valori delle caselle che attraversa.

Esempio 1. Supponete che il campo minato (5x5) sia il seguente:

0 0 1 0 1

0 2 1 0 2

1 0 0 2 0

0 2 0 0 1

0 0 1 0 1

allora ci sono 2 cammini safe di valore massimo che è 7. C'è anche un cammino con valore 6. In caso di diversi cammini con lo stesso valore, come in questo esempio, vogliamo produrre quello più a sinistra.

Si chiede di scrivere una funzione ricorsiva `searchPath` che trovi il cammino di valore massimo (più a sinistra) e lo restituisca come lista concatenata di nodi che, oltre al campo colonna, usato nell'esercizio 1 del 28/4, possiede anche il campo `val` la cui funzione è spiegata nel seguito. La nuova struttura `nodo` è data nel file specificato nell'esercizio. Nell'esercizio si chiede di produrre liste che soddisfano la seguente proprietà: il campo `val` di ogni nodo contiene il valore del cammino safe rappresentato dalla lista che inizia nella casella del campo minato rappresentata da quel nodo. L'Esempio 2 illustra questa condizione.

Esempio 2. La lista a massimo valore da produrre per l'Esempio 1 è la seguente:

[2,7]->[1,6]->[0,4]->[1,3]->[2,1]

dove per ogni nodo rappresentiamo il campo colonna e `val` tra parentesi quadrate, mentre la freccia rappresenta il campo `next`.

La funzione deve soddisfare i seguenti requisiti:

PRE=(X ha $n*n$ valori tutti ≥ 0 , $0 \leq r \leq n-1$, $-1 \leq c \leq n$)

`nodo* searchPath(int * X, int r, int c, int n)`

POST=(se esiste un cammino safe che dalla riga 0 arrivi alla riga $n-1$, allora la funzione restituisce la lista concatenata che rappresenta il cammino a valore massimo con la proprietà spiegata nell'Esempio 2, altrimenti, se non esiste alcun cammino safe che attraversi il campo minato, la funzione restituisce 0)

Correttezza: scrivere la prova induttiva della correttezza della funzione `searchPath` rispetto a PRE e POST date.