Commento al Laboratorio n. 3

Esercizio n.1: **Elemento maggioritario**

Si segue un classico approccio *divide et impera* di tipo *divide and conquer* con a=2 e b=2, quindi con 2 sottoproblemi (vettore sinistro e vettore destro), ciascuno di ampiezza metà. La condizione di terminazione si raggiunge per sottovettori unitari, dove l'elemento maggioritario è chiaramente l'unico elemento del vettore. La ricorsione scende sui 2 sottovettori sinistro e destro. Se l'elemento maggioritario ritornato da entrambi è lo stesso, questo è anche il valore di ritorno della chiamata ricorsiva. Con una scansione di costo lineare si determina il numero di occorrenze nei 2 sotto vettori di ciascuno dei 2 elementi maggioritari ritornati dalle 2 chiamate ricorsive. Se uno di questi risultati supera la metà più uno del numero di elementi del sottovettore, allora si è identificato l'elemento maggioritario del sottovettore e lo sitorna. Se per nessuno dei due risultati vale questa condizione, l'elemento maggioritario non esiste.

Equazione alle ricorrenze:

$$T(n) = 2T(n/2) + n \qquad n>1$$
$$T(1) = 1 \qquad n=1$$

con soluzione $T(n) = O(n\log n)$

Nota 1: si poteva in alternativa ordinare il vettore e poi contare il numero di occorrenze dei suoi elementi. Per soddisfare però i vincoli di complessità e di algoritmo in loco, l'unico ordinamento accettabile era il *quicksort*, nell'assunzione di considerare la complessità di caso medio e non di caso peggiore.

Nota 2: dalla letteratura è noto l'algoritmo di majority vote di Boyer-Moore. Esso è iterativo, ha complessità lineare e non richiede locazioni di memoria aggiuntive in numero dipendente dai dati (è in loco) e quindi soddisfa i vincoli.

Questo esercizio però ha come scopo impratichirsi nella risoluzione ricorsiva di problemi, per cui è auspicabile non ricorrere alle soluzioni delle note 1 e 2.

Esercizio n.2: **Playlist**

Si tratta di una semplice applicazione del principio di moltiplicazione. Essendo le scelte su dati semplici (stringhe per i titoli delle canzoni), la `struct amico`, oltre al numero di canzoni proposte, contiene un vettore di stringhe per i titoli, senza quindi ricorrere agli interi per rappresentare le scelte. La funzione wrapper `princMolt` alloca il vettore delle soluzioni (interi che servono da indici per il vettore di stringhe dei titoli) e la funzione del principio di moltiplicazione. La funzione `princMoltR` è standard.