

Esercitazione 5: strutture dinamiche a nodi concatenati

Esercizio 1. Aggiungere alla classe `PilaDinamica` vista a lezione un metodo `toArray` che ritorna un array con tutti gli elementi della pila, nell'ordine in cui compaiono.

Esercizio 2. Implementare i seguenti metodi in modo iterativo o ricorsivo, a scelta. Nota: tali metodi non realizzano una operazione di una specifica struttura dati (come `PilaDinamica` o `MiniLinkedList`), ma si limitano a manipolare (o produrre) liste concatenate di nodi. Possono essere definiti nella stessa classe di prova che contiene il metodo `main`.

1. `public static Node fromTo(int m, int n)`
ritorna una lista contenente tutti i numeri interi compresi tra `m` ed `n`, estremi inclusi. Per esempio, `fromTo(-1, 2)` deve ritornare la lista `[-1, 0, 1, 2]` mentre `fromTo(2, -1)` deve ritornare la lista vuota `[]`.
2. `public static int occurrences(Node p, int x)`
ritorna il numero di occorrenze di `x` in `p`. Per esempio, se `p` rappresenta la lista `[3, 1, 2, 1, 0]` e `x` è `1`, il metodo deve ritornare `2`.
3. `public static boolean included(Node p, Node q)`
ritorna `true` se tutti gli elementi nella lista `p` compaiono nello stesso ordine anche nella lista `q`, e `false` altrimenti. Per esempio, se `p` e `q` rappresentano rispettivamente le liste `[1, 2, 3]` e `[0, 1, 2, 2, 0, 3, 4]` il metodo deve ritornare `true`.
4. `public static Node reverse(Node p)` ☠️
ritorna la lista che contiene gli stessi elementi di `p`, ma in ordine inverso. Per esempio, se `p` rappresenta la lista `[1, 2, 3]` il metodo deve ritornare la lista `[3, 2, 1]`. Attenzione: la versione ricorsiva è più difficile di quella iterativa.
5. `public static Node zipSum(Node p, Node q)` ☠️
ritorna la lista con le somme degli elementi corrispondenti di `p` e `q`, che si assume abbiano la stessa lunghezza. Per esempio, se `p` e `q` rappresentano rispettivamente le liste `[1, 2, 3]` e `[4, 5, 6]` allora la lista risultante deve essere `[5, 7, 9]`. Attenzione: la versione ricorsiva è più facile di quella iterativa.

Tali metodi **non devono modificare** in alcun modo le liste prese in ingresso come parametri. In particolare, i metodi `reverse` e `zipSum` devono ritornare una **nuova** lista.