

Cognome e Nome:  
Numero di Matricola:

Docente:

**Spazio riservato alla correzione**

1	2	3	4	5	totale
/10	/10	/20	/20	/30	/90

1. **[10 punti]** Si aggiunga alla classe **ArrayStack** (implementa l'interfaccia **Stack** usando un array) il metodo **Stack clone()** che restituisce un nuovo stack avente lo stesso contenuto dello stack su cui è invocato. Il metodo deve lasciare inalterato il contenuto dello stack su cui è invocato. Quale è la complessità del metodo proposto (giustificare la risposta).
2. **[10 punti]** Sia **ArrayQueue** la classe che implementa l'interfaccia **Queue** mediante un array. Aggiungere il metodo **void reverse()** che inverte il contenuto della coda.
3. **[20 punti]** Sviluppare la classe **VectorMap** che implementa l'interfaccia **Map** utilizzando come variabile d'istanza un vettore.
4. **[20 punti]** Aggiungere alla classe **LinkedTree** il metodo **void makeSameArity(int k)** che, ricevuto un intero  $k$ , modifica l'albero su cui è invocato in modo che ogni nodo interno abbia  $k$  figli (tutti i nodi interni devono avere la stessa arità). L'albero modificato è ottenuto aggiungendo (togliendo) ad ogni nodo interno con  $f < k$  ( $f > k$ ) figli,  $k-f$  ( $f-k$ ) figli. I figli aggiunti saranno delle foglie aventi come elemento la stringa "new". Valutare la complessità del metodo proposto.
5. **[30 punti]** Scrivere lo pseudocodice dell'algoritmo per la visita in ampiezza di un grafo (BFS). Commentare l'uso delle strutture dati utilizzate. [10 punti]

Indicare, giustificando la risposta, la complessità dell'algoritmo BFS. [5 punti]

Codificare l'algoritmo BFS in Java (scrivere una funzione Java che implementi l'algoritmo). Il codice deve far riferimento alle interfacce illustrate durante il corso. [15 punti].