Algoritmi e Strutture Dati
Esercitazione 5 (Laboratorio)

Esercizio 1

Implementare in C la funzione

- `void insertionSort(float* a, int n)`

che, effettuando side-effect, ordina il vettore di input `a`, di `n` reali, in ordine decrescente, eseguendo l'algoritmo *InsertionSort*.

Esercizio 2

Implementare in C la funzione

- `int* mergeSort(int* a, int n)`

che, preso in input un vettore `a` di `n` interi, restituisce un nuovo vettore contenente gli elementi di `a` in ordine crescente, ordinati applicando l'algoritmo *MergeSort*.

Esercizio 3

1) Definire lo pseudocodice dei seguenti algoritmi:

1. $profondita(Albero\ a) \rightarrow intero$
   restituisce la profondità dell'albero di input a;
2. $quantiNodi(Albero\ a) \rightarrow intero$
   restituisce il numero di nodi dell'albero di input a;
3. $quanteFoglie(Albero\ a) \rightarrow intero$
   restituisce il numero di foglie dell'albero di input a.

2) Valutare la complessità di ciascuno degli algoritmi sopra definiti.

3) Assumendo di usare una rappresentazione collegata, implementare gli algoritmi definiti al punto 1) nelle seguenti funzioni C:

1. `int profondita(Albero a);`
2. `int quantiNodi(Albero a);`
3. `int quanteFoglie(Albero a).`