Laboratorio di Algoritmi e Strutture dati*

6 Giugno 2015

Si implementi una versione della struttura dati UnionFind come descritto dai seguenti esercizi. Si scelga una sola tra le due implementazioni proposte.

Esercizio UnionFind – *(6pt)*

Si definisca una classe `UnionFind` (come suggerito nel codice in basso) che risolve il problema `union-find` visto a lezione. Gli elementi degli insiemi sono di tipo `int` e variano in un range $[0 \dots n-1]$; il valore di n viene passato al costruttore della classe UnionFind.

```
public class UnionFind {
  /*
    Crea la struttura dati e imposta la capacità a n
   */
  UnionFind(int n) { ... }

  /*
    restituisce la capacità
   */
  int getCapacity() { ... }

  /*
    cambia la capacità; se newN < getCapacity() solleva una
    eccezione
   */
  void setCapacity(int newN) throws IllegalArgumentException {
    ...
  }

  /*
    restituisce il rappresentante dell'insieme in cui si trova e
   */
  int find(int e) { ... }

  /*
    Kruskal-Union

    Unisce l'insieme che contiene a e l'insieme che contiene b.
```

*Revisione n. 1.0 del 11 maggio 2015

```
    Se a e b fanno parte dello stesso insieme restituisce false e
    non fa nulla, altrimenti unisce gli insiemi di cui a e b fanno
    parte e restituisce true.
   */
  boolean union(int a, int b) { ... }
}
```

Esercizio UnionFind Generica – *(10pt)*

Si definisca una classe generica `UnionFind` (come suggerito nel codice in basso) che risolve il problema `union-find` visto a lezione. Gli elementi degli insiemi sono di tipo generico T.

```
public class UnionFind<T> {
  /*
    Crea la struttura dati
   */
  UnionFind() { ... }

  /*
    Crea un nuovo set contenente solo l'elemento T, solleva una
    eccezione in caso elem sia già presente
   */
  void makeSet(T elem) { ... }

  /*
    restituisce il rappresentante dell'insieme in cui si trova e
   */
  T find(T e) { ... }

  /*
    Kruskal-Union

    Unisce l'insieme che contiene a e l'insieme che contiene b.

    Se a e b fanno parte dello stesso insieme restituisce false e
    non fa nulla,
    altrimenti unisce gli insiemi di cui a e b fanno parte e
    restituisce true.
  */
  boolean union(T a, T b) { ... }
}
```