

Tema 6 - Ejercicios Resueltos

Estructuras de datos y algoritmos (Universitat Politecnica de Valencia)

MF-Sets

EJERCICIOS RESUELTOS

Ejercicio 1.- Diseña un método en la clase ForestMFSet que devuelva cuántos elementos tiene la clase (conjunto) al que pertenece un elemento dado a.

Solución:

```
public int cardinal(int a) {
  int raiz = find(a), n = 0;
  for (int i=0; i < talla; i++) if (find(i)==raiz) n++;
  return n:
}
```

Ejercicio 2.- Diseña un método en la clase ForestMFSet que devuelva una ListaConPI con los identificadores de cada una de las clses (conjuntos) que hay en un MFSet.

Solución:

```
public ListaConPI<Integer> identificadores() {
  ListaConPI<Integer> res = new LEGListaConPI<Integer>();
  for ( int i=0; i < talla; i++ )
    if ( elArray[i] < 0 ) res.insertar(elArray[i]);</pre>
  return res;
}
```

Ejercicio 3.- Diseña un método en la clase *ForestMFSet* que devuelva el número de clases que hay en un MFSet. Indica también qué se podría hacer para que este método tenga un coste independiente de la talla del problema.

Solución:

```
public int numeroClases(){
  int res = 0;
  for ( int i=0; i<talla; i++ ) if ( elArray[i]<0 ) res++;
  return res;
}
```

Ejercicio 4.- Diseña un método en la clase ForestMFSet que devuelva una ListaConPI con los elementos de la clase (conjunto) a la que pertenece un elemento dado a.

Solución:

```
public void elementosEnClase(int a) {
  ListaConPI<Integer> res = new LEGListaConPI<Integer>();
  int raiz = find(a);
  for ( int i=0; i<talla; i++) if ( find(i)==raiz ) res.insertar(i);</pre>
  return res;
}
```

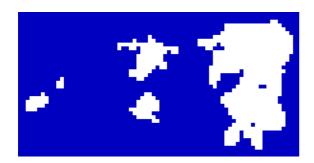


Ejercicio 3.- Diseña un método en la clase *ForestMFSet* que devuelva el número de elementos que habría en la clase (conjunto) resultante de unir las clases (conjuntos) a las que pertenecen dos elementos dados *a* y *b*.

Solución:

```
public void tallaUnion(int a, int b) {
  int raizA = find(a), raizB = find(b), n = 0;
  for ( int i=0; i<talla; i++ ) {
    int aux = find(i);
    if ( aux==raizA || aux==raizB ) n++;
  }
  return n;
}</pre>
```

Ejercicio 4.- Se dispone de una matriz de tipo *boolean* que representa un mapa de un archipiélago. Un valor true en una posición (x,y) indica que hay tierra en ese punto, mientras que un valor false indica que hay mar.



Ejemplo:

Los valores *true* y *false* se representan mediante los colores blanco y azul, respectivamente.

Implementa un método con el siguiente perfil que permita calcular el número de islas que hay en el archipiélago: public static int numIslas(boolean[][] mapa);

Solución:

```
public static int numIslas(boolean[][] mapa) {
  int ancho = mapa.length, alto = mapa[0].length, res = 0;
  MFSet m = new ForestMFSet(ancho * alto);
  // Crear las clases de equivalencia
  for ( int i=0; i<ancho; i++ )</pre>
    for (int j=0; j<alto; j++ ) {</pre>
      if ( i+1<ancho && mapa[i][j]==mapa[i+1][j )</pre>
       m.merge(j*ancho+i, j*ancho+i+1);
      if ( j+1<alto && mapa[i][j]==mapa[i][j+1] )</pre>
        m.merge(j*ancho+i, (j+1)*ancho+i);
  }
  // Calcular el número de islas (clases de tierra)
 boolean[] visitado = new boolean[ancho*alto];
  for ( int i=0; i<ancho; i++ )</pre>
    for ( int j=0; j<alto; j++ )
      if ( mapa[i][j] ) {
                           // Tierra
        int raiz = m.find(j*ancho+i);
        if ( !visitado[raiz] ) { visitado[raiz] = true; res++; }
      }
  return res;
}
```