

# Praktikum DAA: Disjoint Sets

## Disjoint Sets

Buatlah sebuah kelas bernama DisjointSets. Kelas ini mengimplementasikan Disjoint-Sets dengan Rooted Tree menggunakan Array. Lengkapi operasi **FIND-SET**, dan **UNION**. Gunakan heuristic *path-compression* pada operasi FIND-SET dan *union-by-rank* pada operasi UNION (jika rank sama, maka representatif element2 menjadi root dari tree hasil UNION). Pergunakan dan memodifikasi kerangka kelas di bawah ini. Perhatikan bahwa baris 34-35 merupakan operasi **MAKE-SET**.

```
1  import java.util.Scanner;
2  import java.util.*;
3
4  /**
5   * <h1>Kelas Disjoint Sets</h1>
6   * implementasi dari disjoint sets dengan array.
7   * Prosedur findsets dengan path-compression dan prosedur union dengan union-by-rank.
8   */
9  public class DisjointSets {
10     /**
11      * jumlah elemen pada disjoint sets
12      */
13     int numElements; //jumlah elemen pada disjoint sets
14     /**
15      * menyimpan parent dari setiap elemen
16      */
17     int[] parents;
18     /**
19      * menyimpan rank dari setiap elemen
20      */
21     int[] ranks;
22
23     /**
24      * Constructor dari kelas DisjointSets. Jumlah elemen tidak
25      * dapat berubah. Versi yang lebih baik bisa berubah (gunakan vector).
26      * Elemen adalah bilangan bulat.
27      * @param numElements jumlah elemen dari disjoint sets.
28      */
29     DisjointSets(int numElements) {
30         this.numElements = numElements;
31         this.parents = new int[numElements];
32         this.ranks = new int[numElements];
33         for (int i=0;i<this.numElements;i++) {
34             this.parents[i]=i; //parent dari setiap elemen adalah elemen itu sendiri
35             this.ranks[i]=0; //rank dari setiap elemen adalah 0, tinggi pohon dengan root saja = 0
36         }
37     }
38
39     /**
40      * Method findSets mengembalikan representatif dari set di mana element berada.
41      * Path-compression diimplementasikan saat melakukan pencarian elemen
42      * @param element Elemen yang hendak dicari representatifnya
43      * @return elemen representative dan
44      */
45     public int findSets(int element) {
46
47     }
48
49     /**
50      * Method union menggabungkan sets di mana element1 dan element2 berada.
51      * Union-by-rank diimplementasikan saat menggabungkan kedua sets (keduanya bisa saja sama)
52      * @param element1 elemen pertama
53      * @param element2 elemen kedua
54      */
55
56     public void union(int element1,int element2) {
57
58     }
59
60     /**
61      * Main program untuk mencoba kelas DisjointSets
62      */
63     public static void main(String[] args) {
64         Scanner sc = new Scanner(System.in);
65
66     }
67 }
68
```

Untuk menguji implementasi Anda, gunakan method main untuk menerima input dan mengeluarkan output seperti yang diminta pada soal berikut:

## Uji Kelas DS

Buatlah sebuah program yang menguji kelas disjoint sets yang telah dibuat sebelumnya.

### Spesifikasi Masukan

Input diawali dengan bilangan bulat  $n$  ( $1 \leq n \leq 1024$ ), yaitu banyaknya elemen pada disjoint sets. Selanjutnya, setiap baris masukan merepresentasikan satu operasi yang harus dilakukan:

- $f \ E$  :melakukan operasi find set, yaitu mencari representatif dari suatu elemen  $E$ .
- $u \ E1 \ E2$  :melakukan operasi union, yaitu menggabungkan set dari elemen  $E1$  dan set dari elemen  $E2$
- $p \ E$  :menampilkan parent dari elemen  $E$
- $r \ E$  :menampilkan rank dari elemen  $E$
- $t$  :menampilkan parent dari seluruh elemen
- $r$  :menampilkan rank dari seluruh elemen
- $x$  :akhir masukan

### Spesifikasi Keluaran

Baris pertama dari keluaran selalu: `make-set n times` dengan  $n$  adalah banyaknya elemen pada disjoint sets. Selanjutnya, keluaran disesuaikan dengan operasi yang harus dilakukan. Spesifikasi keluaran dari setiap operasi dapat dilihat di contoh keluaran. Untuk menampilkan isi dari array parent dan rank, gunakan perintah `Arrays.toString()`

### Teladan Masukan

```
6
f 0
f 2
u 0 2
t
k
p 0
r 0
p 2
r 2
u 0 1
t
k
u 5 3
p 5
r 5
p 3
r 3
u 5 4
p 4
r 4
t
k
u 0 3
t
k
f 0
t
k
x 2
```

## Teladan Keluaran

```
make-set 6 times
representative of 0 is 0
representative of 2 is 2
union between 0 and 2
list of parents: [2, 1, 2, 3, 4, 5]
list of ranks   : [0, 0, 1, 0, 0, 0]
parents of 0 is 2
ranks of 0 is 0
parents of 2 is 2
ranks of 2 is 1
union between 0 and 1
list of parents: [2, 2, 2, 3, 4, 5]
list of ranks   : [0, 0, 1, 0, 0, 0]
union between 5 and 3
parents of 5 is 3
ranks of 5 is 0
parents of 3 is 3
ranks of 3 is 1
union between 5 and 4
parents of 4 is 3
ranks of 4 is 0
list of parents: [2, 2, 2, 3, 3, 3]
list of ranks   : [0, 0, 1, 1, 0, 0]
union between 0 and 3
list of parents: [2, 2, 3, 3, 3, 3]
list of ranks   : [0, 0, 1, 2, 0, 0]
representative of 0 is 3
list of parents: [3, 2, 3, 3, 3, 3]
list of ranks   : [0, 0, 1, 2, 0, 0]
```