Nama : Andyan Yogawardhana

NIM : 21/482180/PA/21030

Kelas : KOMB1

Tugas 9 – Disjoint Set

Source Code

1. public class Disjoint {
2. public static void main(String[] args) {
3. DisjointSet disjointSet = new DisjointSet(5);
4. System.out.println("\nInitial set\n");
5. disjointSet.printResult();
6. disjointSet.union(3, 4);
7. System.out.println("\nAfter union 3 and 4\n");
8. disjointSet.printResult();
10. disjointSet.union(1, 2);
11. disjointSet.union(1, 3);
12. System.out.println("\nFinal result\n");
13. disjointSet.printResult();
14. }
15. }
16. class Set {
17. private int parent, rank;
18. public Set(int data) {
19. this.parent = data;
20. this.rank = 0;
21. }
22. public int getParent() {
23. return this.parent;
24. }
25. public void setParent(int data) {
26. this.parent = data;
27. }
28. public int getRank() {
29. return this.rank;
30. }
31. public void setRank(int data) {
32. this.rank = data;
33. }
34. }
35. class DisjointSet {
36. private Set[] sets;
37. private int[] elements;     *// menyimpan jumlah anggota elemen masing-masing set*
38. private int size, setCount; *// setCount: menyimpan jumlah set yang terbentuk*
39. public DisjointSet(int numItem) {
40. this.size = numItem;
41. this.setCount = numItem;    *// jumlah set awal yang terbentuk sama dengan jumlah elemen awal*
42. this.sets = new Set[size + 1];
43. this.elements = new int[size + 1];
44. for(int i = 1; i <= this.size; i++) {
45. this.sets[i] = new Set(i);
46. this.elements[i] = 1;   *// setiap set baru yang terbentuk memiliki satu buah elemen*
47. }
48. }
49. public int find(int item) {
50. int parent = this.sets[item].getParent();
52. if(item == parent) {
53. return item;
54. }
55. else {
56. parent = find(parent);
57. this.sets[item].setParent(parent); *// path compression*
58. return parent;
59. }
60. }
61. public boolean isSameSet(int firstItem, int secondItem) {
62. return find(firstItem) == find(secondItem);
63. }
64. public void union(int firstItem, int secondItem) {
65. int firstItemParent = find(firstItem);
66. int secondItemParent = find(secondItem);
67. if(firstItemParent != secondItemParent) {
68. int firstRank = this.sets[firstItemParent].getRank();
69. int secondRank = this.sets[secondItemParent].getRank();
70. if(firstRank < secondRank) {
71. this.sets[firstItemParent].setParent(secondItemParent);
72. this.elements[firstItemParent] += this.elements[secondItemParent];  *// menambahkan jumlah elemen pada set yang menjadi root dengan jumlah elemen dari set yang bergabung*
73. }
74. else if (firstRank > secondRank) {
75. this.sets[secondItemParent].setParent(firstItemParent);
76. this.elements[secondItemParent] += this.elements[firstItemParent];  *// menambahkan jumlah elemen pada set yang menjadi root dengan jumlah elemen dari set yang bergabung*
77. }
78. else {
79. this.sets[secondItemParent].setParent(firstItemParent);
80. this.sets[firstItemParent].setRank(firstRank + 1);
81. this.elements[firstItemParent] += this.elements[secondItemParent];  *// menambahkan jumlah elemen pada set yang menjadi root dengan jumlah elemen dari set yang bergabung*
82. this.elements[secondItemParent] = 0;    *// set yang bergabung ke set lain tidak memiliki elemen lagi setelah dilakukan union*
83. }
84. setCount--;     *// jumlah tree yang terbentuk berkurang setelah dua buah set bergabung menjadi satu*
85. }
86. }
87. public void printResult() {
88. print();
89. printRank();
90. countElement();
91. countSet();
92. System.out.println("----------------------------");
93. }
94. public void print() {
95. for(int i = 1; i <= this.size; i++) {
96. System.out.println("- Parent of " + i + " = " + find(i));
97. }
98. }
99. public void printRank() {
100. for(int i = 1; i <= this.size; i++) {
101. System.out.println("> Rank of " + i + " = " + this.sets[i].getRank());
102. }
103. }
105. public void countElement() {
106. for(int i = 1; i <= size; i++) {
107. System.out.println("- Set " + i + " has " + elements[i] + " element(s)");
108. }
109. }
111. public void countSet() {
112. System.out.println("> Total sets created = " + setCount + "\n");
113. }
114. }

Output Terminal

