**Практическая работа №13**

**Вариант №1 – Основные алгоритмы работы с графами.**

**Автор: Николаев-Аксенов И. С.**

**Группа: ИКБО-20-19**

**Задание:**

Составить программу реализации алгоритма Крускала построения остовного дерева минимального веса.

Выбрать и реализовать способ представления графа в памяти.

Предусмотреть ввод с клавиатуры произвольного графа.

Разработать доступный способ (форму) вывода результирующего дерева на экран монитора.

Провести тестовый прогон программы для заданного графа в соответствии с индивидуальным заданием

**Код программы:**

*Graph.java:*

1. **public** **class** Graph **implements** Comparable<Graph> {
2. **public** **int** A;
3. **public** **int** mass;
4. **public** **int** B;
6. Graph(**int** A, **int** mass, **int** B) {
7. **this**.A=A;
8. **this**.B=B;
9. **this**.mass=mass;
10. }
12. @Override
13. **public** **int** compareTo(Graph o) {
14. **if**(mass!=o.mass){
15. **return** mass<o.mass? -1:1;
16. }
17. **return** 0;
18. }
19. }

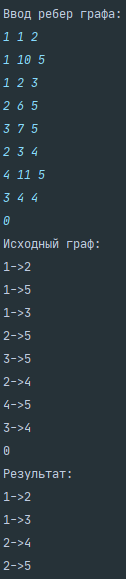
*SetGraph.java:*

1. **public** **class** SetGraph {
2. **int**[] number, rang;
4. SetGraph(**int** size) {
5. number=**new** **int**[size];
6. rang=**new** **int**[size];
7. **for**(**int** i=0;i<size;++i){
8. number[i]=i;
9. }
10. }
12. **int** set(**int** x) {
13. **return** x==number[x]? x:(number[x]=set(number[x]));
14. }
16. **boolean** merge(**int** A, **int** B){
17. **if**(set(A)==set(B))
18. **return** **false**;
20. **if**(rang[A]<rang[B])
21. number[A]=B;
23. **else** {
24. number[B]=A;
25. **if**(rang[A]==rang[B])
26. rang[A]++;
27. }
28. **return** **true**;
29. }
30. }

*StartKruskalsAlgorithm.java:*

1. **import** java.util.\*;
3. **public** **class** StartKruskalsAlgorithm {
4. **public** **static** **int** KruskalAlgorithm(ArrayList<Graph> graph) {
5. SetGraph union = **new** SetGraph(graph.size()+1);
6. Collections.sort(graph);
8. ArrayList<Graph> buff = **new** ArrayList<>();
9. **for**(Graph i: graph) {
10. **if**(union.merge(i.A,i.B)){
11. buff.add(i);
12. }
13. }
14. graph.clear();
15. graph.addAll(buff);
16. **return** 0;
17. }
19. **public** **static** **void** main(String[] args) {
20. ArrayList<Graph> graph = **new** ArrayList<>();
21. Scanner scanner = **new** Scanner(System.in);
22. **int** a,b,m;
24. System.out.println("Ввод ребер графа: ");
25. **while**(**true**){
26. a = scanner.nextInt();
27. **if**(a==0) **break**;
28. m = scanner.nextInt();
29. b = scanner.nextInt();
30. graph.add(**new** Graph(a,m,b));
31. }
32. System.out.println("Исходный граф: ");
33. **for** (Graph item : graph) {
34. System.out.println(item.A + "->" + item.B);
35. }
36. System.out.println(KruskalAlgorithm(graph));
37. System.out.println("Результат: ");
38. **for** (Graph value : graph) {
39. System.out.println(value.A + "->" + value.B);
40. }
41. System.out.println("В данной работе используется язык graphviz. Посмотреть граф можно на сайте graphvix.org.");
42. }
43. }

**Результат выполнения программы:**



|  |  |
| --- | --- |
| *Исходный граф:* | *Результат:* |
|  |  |