Assignment 3

Design and analysis of algorithms

By Dmitriy Belyaikin and Danial Myrzatayev

Input data


```
{
  "graphs": [
    {
      "id": 0,
      "vertices": ["A", "B", "C", "D", "E"],
      "edges": [
        {"from": "A", "to": "B", "weight": 4},
        {"from": "A", "to": "C", "weight": 3},
        {"from": "B", "to": "C", "weight": 2},
        {"from": "B", "to": "D", "weight": 5},
        {"from": "C", "to": "D", "weight": 7},
        {"from": "C", "to": "E", "weight": 8},
        {"from": "D", "to": "E", "weight": 6}
      ]
    },
    {
      "id": 1,
      "vertices": ["A", "B", "C", "D"],
      "edges": [
        {"from": "A", "to": "B", "weight": 1},
        {"from": "A", "to": "C", "weight": 4},
        {"from": "B", "to": "C", "weight": 2},
        {"from": "C", "to": "D", "weight": 3},
        {"from": "B", "to": "D", "weight": 5}
      ]
    }
  ]
}
```

Results

(JSON file was cleaned up and formatted for readability)


```
{
  "0": {
    "prims": {
      "ExecutionResults": {
        "operations": 82,
        "executionTime": 3150000
      }
    },
    "kruskals": {
      "ExecutionResults": {
        "operations": 34,
        "executionTime": 2039700
      }
    }
  },
  "1": {
    "prims": {
      "ExecutionResults": {
        "operations": 53,
        "executionTime": 107700
      }
    },
    "kruskals": {
      "ExecutionResults": {
        "operations": 24,
        "executionTime": 43300
      }
    }
  }
}
```

Results

Broken ↘

```
for (Graph<String> graph : graphs) {
    ExecutionResults primsResults = new Prim<>(graph).run();
    ExecutionResults kruskalsResults = new Kruskal<>(graph).run();

    Map<String, ExecutionResults> results = new HashMap<>();
    results.put("prims", primsResults);
    results.put("kruskals", kruskalsResults);

    resultsMap.put(graph.id(), results);
}
```

Fixed ↘

```
for (Graph<String> graph : graphs) {
    Graph<String> graphCopy = graph;
    ExecutionResults primsResults = new Prim<>(graph).run();
    ExecutionResults kruskalsResults = new Kruskal<>(graphCopy).run();

    Map<String, ExecutionResults> results = new HashMap<>();
    results.put("prims", primsResults);
    results.put("kruskals", kruskalsResults);

    resultsMap.put(graph.id(), results);
}
```

- **Prim's algorithm required almost a double of Kruskal's operations and time to run** (Dmitriy's implementation).
- **Input data was identical.**
- The possible faulty results code was fixed, but the difference wasn't big.

- https://github.com/belyaikin/prim-kruskal-argorithms