Disjoint Sets Data Structures and Union Find Algorithm

Last updated @ 29 October 2017
First published @ 23 September 2017

What are Disjoint Sets?

Disjoint sets are sets whose intersection is the empty set. For example, {0, 1, 2} and {3} are disjoint sets, while {0, 1, 2} and {2, 3} are not.

Problem

Given some separated elements grouped in disjoint sets. **Find** whether 2 elements are in the same set or not.

Approach

- **How to determine if 2 elements are in the same set?** Check if they have the same set root/representative.

- **How to determine the root/representative of a set?** Choose the biggest index of a set or by rank.

Implementation

Find with compress path and Union by biggest index

src/com/hellokoding/core/disjoinset/DisjoinSetUnionByBiggestIndex.java

```
package com.hellokoding.core.disjoinset;
```

```java
public class DisjoinSetUnionByBiggestIndex {

    int[] parents;
    int N;

    DisjoinSetUnionByBiggestIndex(int N) {
        this.N = N;
        parents = new int[N];
        initDisjoinSet();
    }

    void initDisjoinSet() {
        for (int i = 0; i < N; i++) {
            parents[i] = i;
        }
    }

    int findRoot(int x) {
        if (parents[x] != x) {
            parents[x] = findRoot(parents[x]);
        }

        return parents[x];
    }

    void union(int x, int y) {
        int rootOfX = findRoot(x);
        int rootOfY = findRoot(y);

        if (rootOfX == rootOfY) {
            return;
        }

        parents[rootOfX] = rootOfY;
    }

    public static void main(String[] args) {
        int N = 4;
        DisjoinSetUnionByBiggestIndex disjoinSet = new DisjoinSetUnion
            disjoinSet union(0, 1);
```

```
disjoinSet.union(0, 1);
disjoinSet.union(1, 2);


        // Check if 1 is in the same set with 2
        if (disjoinSet.findRoot(1) == disjoinSet.findRoot(2)) {
            System.out.println("1 is in the same set with 2");
        } else {
            System.out.println("1 is not in the same set with 2");
        }

        // Check if 1 is in the same set with 3
        if (disjoinSet.findRoot(1) == disjoinSet.findRoot(3)) {
            System.out.println("1 is in the same set with 3");
        } else {
            System.out.println("1 is not in the same set with 3");
        }
    }
}
```

Inital state of `parents`
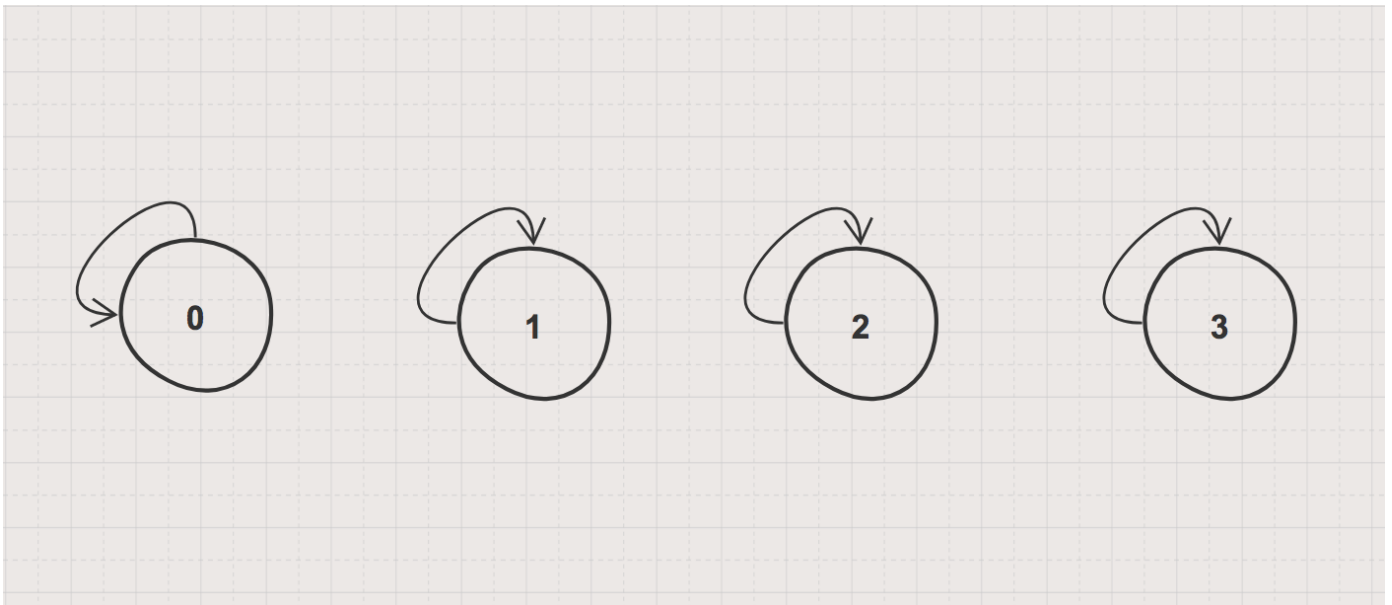
Final state of `parents`
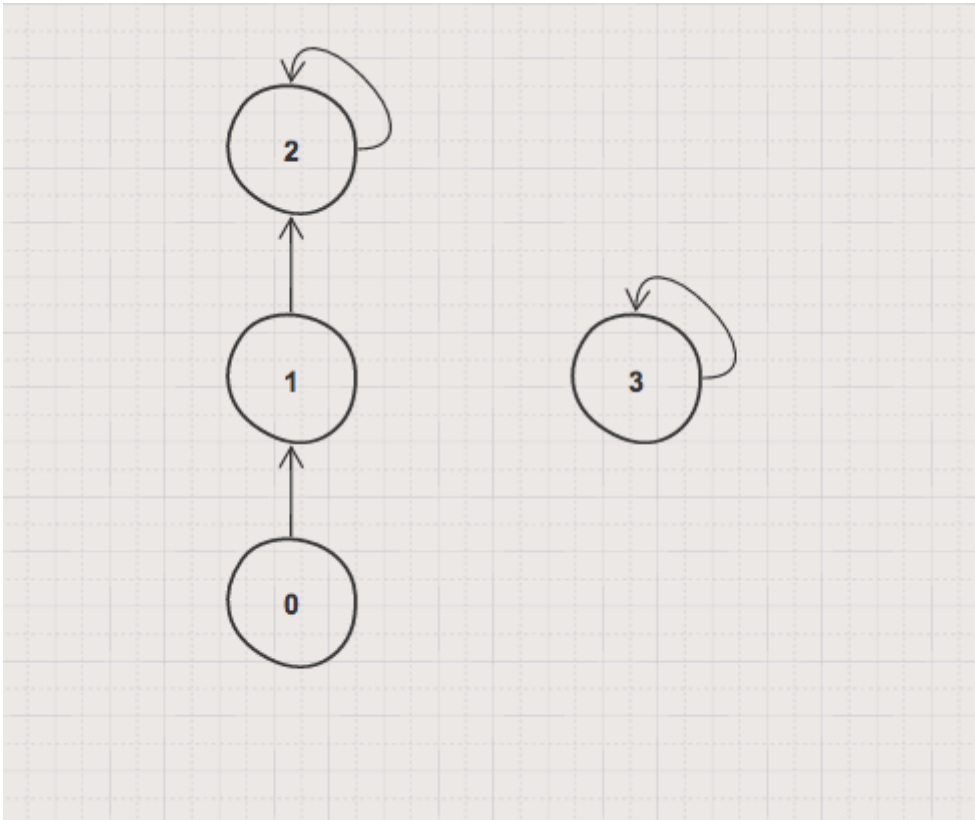
Find with compress path and Union by rank

src/com/hellokoding/core/disjoinset/DisjoinSetUnionByRank.java

```java
package com.hellokoding.core.disjoinset;

public class DisjoinSetUnionByRank {
    int[] parents, ranks;
    int N;

    DisjoinSetUnionByRank(int N) {
        this.N = N;
        parents = new int[N];
        ranks = new int[N];
        initDisjoinSet();
    }

    void initDisjoinSet() {
        for (int i = 0; i < N; i++) {
```

```java
            parents[i] = i;
        }
    }

    int findRoot(int x) {
        if (parents[x] != x) {
            parents[x] = findRoot(parents[x]);
        }

        return parents[x];
    }

    void union(int x, int y) {
        int rootOfX = findRoot(x);
        int rootOfY = findRoot(y);

        if (rootOfX == rootOfY) {
            return;
        }

        if (ranks[rootOfX] < ranks[rootOfY]) {
            parents[rootOfX] = rootOfY;
        } else if (ranks[rootOfX] > ranks[rootOfY]) {
            parents[rootOfY] = rootOfX;
        } else {
            parents[rootOfX] = rootOfY;
            ranks[rootOfY] = ranks[rootOfY] + 1;
        }
    }
}


public static void main(String[] args) {
    int N = 4;
    DisjoinSetUnionByRank disjoinSet = new DisjoinSetUnionByRank(N);
    disjoinSet.union(0, 1);
    disjoinSet.union(1, 2);

    // Check if 1 is in the same set with 2
    if (disjoinSet.findRoot(1) == disjoinSet.findRoot(2)) {
        System.out.println("1 is in the same set with 2");
    } else {
```
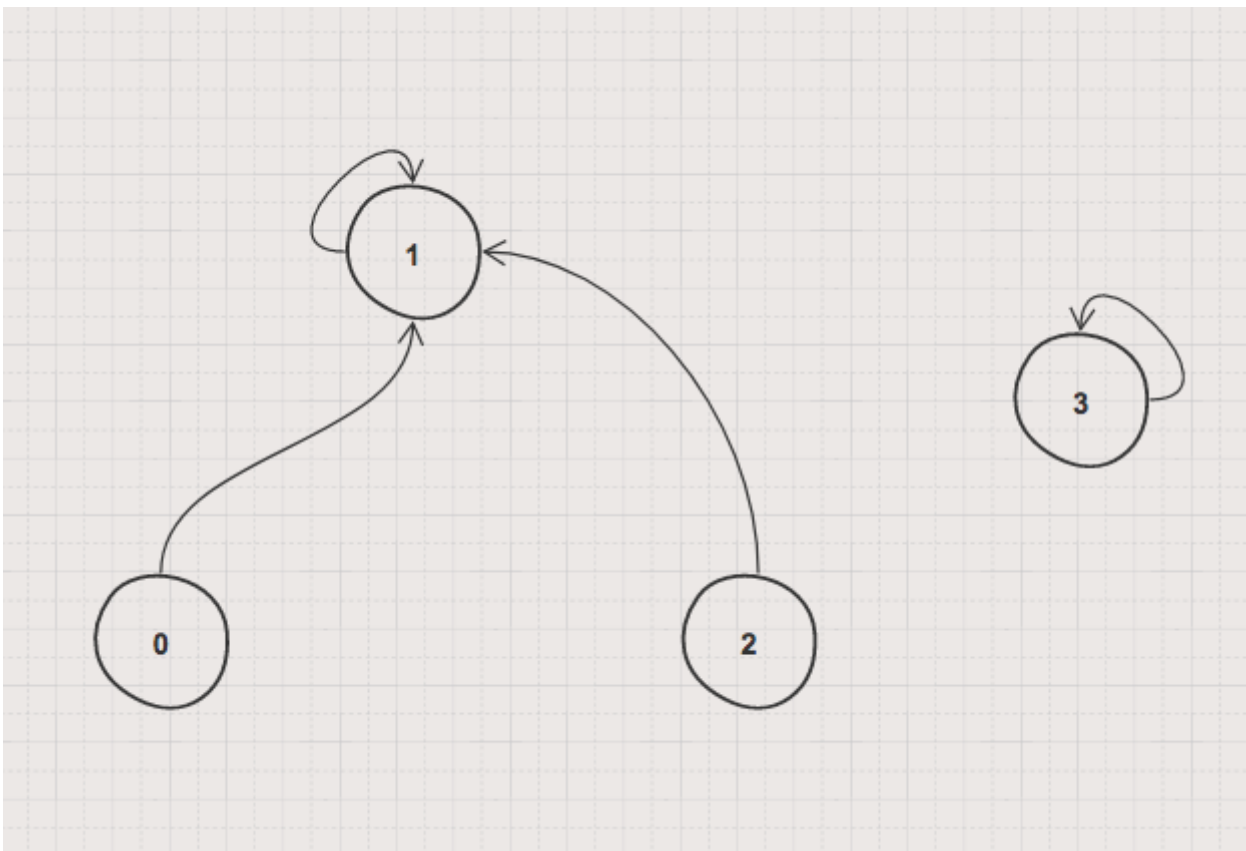
```
    } else {
        System.out.println("1 is not in the same set with 2");
    }

    // Check if 1 is in the same set with 3
    if (disjoinSet.findRoot(1) == disjoinSet.findRoot(3)) {
        System.out.println("1 is in the same set with 3");
    } else {
        System.out.println("1 is not in the same set with 3");
    }
  }
}
```

Final state of `parents`

References

[Disjoin-set Data Structures (topcoder.com)](topcoder.com)

Core   # Data Structure   # DisjoinSet   # Algorithm   # UnionFind   # Java

Giau Ngo

Giau Ngo is a software engineer, creator of Hello Koding. He is interested in making simple code examples and tutorials. Follow him on GitHub/Twitter or drop a message hellokoding [at] gmail [dot] com

🐦 🔗

RESTful API Example with Spring Boot, Spring Data REST, Spring Data JPA Many To Many Extra Columns Relationship and MySQL

Arch   # RESTful   # Spring Data REST   # Spring Data JPA   # Spring Boot   # Java   # MySQL

cURL

Giau Ngo - 10 October 2017

Longest Increasing Subsequence

Core   # Algorithm   # Dynamic Programming   # Java

Comments

Recent Stories

Streaming Data from Kafka to Postgres with Kafka Connect, AVRO, Schema Registry and Python

RESTful APIs Example with Spring Boot, Integration Test with MockMVC, UI Integration with VueJS

Email Verification Example with Spring Boot, MySQL, Docker Compose, Amazon SES

Docker Compose with Spring Boot, MySQL, NGINX

Follow Us

Related Posts

Longest Increasing Subsequence

Email Verification Example with Spring Boot, MySQL, Docker Compose, Amazon SES

Giau Ngo - 09 January 2018

Docker Compose with Spring Boot, MySQL, NGINX

Hello Koding

Auth

Persistence

REST

○ GitHub

f Facebook

🐦 Twitter

G+ Google Plus

Powered by Ghost