

# 并查集

当需要处理的集合信息比较多时, 可以使用"敌人的敌人就是朋友这种策略", 通过扩大par数组, 使用x和x+N来表示x属于不同的两个集合。 [POJ 2010](#)

```
import java.util.Scanner;

class UnionFind {
    int[] par, rank;
    int n;

    public UnionFind(int n) {
        this.n = n;
        init(n);
    }

    public void init(int n) {
        par = new int[n];
        rank = new int[n];
        for (int i = 0; i < n; i++) {
            par[i] = i;
            rank[i] = 0;
        }
    }

    public int find(int x) {
        if (par[x] == x)
            return x;
        return par[x] = find(par[x]);
    }

    public void unite(int x, int y) {
        x = find(x);
        y = find(y);
        if (x == y)
            return;
        if (rank[x] < rank[y]) {
            par[x] = y;
        } else {
            par[x] = y;
            if (rank[x] == rank[y])
                rank[x]++;
        }
    }

    public boolean same(int x, int y) {
        return find(x) == find(y);
    }
}

public class Main {
    public static void main(String[] args) {
```

```
Scanner input = new Scanner(System.in);
int n = input.nextInt(), m = input.nextInt();
UnionFind solution = new UnionFind(n);
for (int i = 0; i < m; i++)
    solution.unite(input.nextInt(), input.nextInt());
int ans = 0;
for (int i = 0; i < n; i++)
    if (solution.par[i] == i)
        ans++;
System.out.print(ans);
}
}
```

测试数据:

input: n m (n为节点数, m为信息条数, 接下来m条信息代表a,b为同一组, 输出共有几种不同组)

10 9

1 2

3 4

5 2

4 6

2 6

8 7

9 7

1 6

2 4

output:

3