

Disjoint Sets

Ulises Méndez Martínez

Algorist Weekly Talks

ulisesmdzmtz@gmail.com

July 1, 2016

Overview

1 Motivation

- New Year Permutation

2 Disjoint Sets

- Union-Find Key Concepts

3 Implementation

B. New Year Permutation

Codeforces Round 500, Problem B

User ainta has a permutation p_1, p_2, \dots, p_n . As the New Year is coming, he wants to make his permutation as pretty as possible.

Permutation a_1, a_2, \dots, a_n is prettier than permutation b_1, b_2, \dots, b_n , if and only if there exists an integer k ($1 \leq k \leq n$) where $a_1 = b_1, a_2 = b_2, \dots, a_{k-1} = b_{k-1}$ and $a_k < b_k$ all holds.

As known, permutation p is so sensitive that it could be only modified by swapping two distinct elements. But swapping two elements is harder than you think. Given an $n \times n$ binary matrix A , user ainta can swap the values of p_i and p_j ($1 \leq i, j \leq n, i \neq j$) if and only if $A_{i,j} = 1$.

Given the permutation p and the matrix A , user ainta wants to know the prettiest permutation that he can obtain.

Examples

Input

7	5
5 2 4 3 6 7 1	4 2 1 5 3
0001001	00100
0000000	00011
0000010	10010
1000001	01101
0000000	01010
0010000	
1001000	

Output

1 2 4 3 6 7 5	1 2 3 4 5
---------------	-----------

Union Find Theory

- Representative
- Disjoint Sets
- Path compression
- Union by Rank

Implementation part I

Example (C++ Implementation)

```
#define MAX 1000000
int p[MAX], rank[MAX];

int find_set(int x){
    if (x != p[x])
        p[x] = find_set(p[x]);
    return p[x];
}

void union_set(int x, int y){
    link(find_set(x), find_set(y));
}
```

Implementation part II

Example (C++ Implementation)

```
void make_set(int x){
    p[x] = x;
    rank[x] = 0;
}

void link(int x, int y){
    if (rank[x] > rank[y])
        p[y] = x;
    else{
        p[x] = y;
        if (rank[x] == rank[y])
            rank[y] = rank[y] + 1;
    }
}
```

References

- Argentina Training Camp 2012: <https://goo.gl/iesQFX>
- Wikipedia: <https://goo.gl/3TuSCI>
- TopCoder Tutorial: <https://goo.gl/O8pi6C>
- Hackerearth Notes: <https://goo.gl/TmDEYH>
- Visualgo site: <http://visualgo.net/ufds>