Система непретинних множин

СТУДЕНТ ГРУПИ ФВЕ

КІРГЕТОВ ДАНІІЛ

Система неперетинних множин

Структура даних, що дозволяє проводити роботу із множиною елементів, що розбиті на неперетинні підмножини

Система неперетинних множин

Отже, нехай маємо систему із декількох множин. При цьому жодна із цих підмножин не перетинається із будь-якою іншою.

Для реалізації СНМ задаються 3 операції і кожній підмножині надається деякий представник.

Операції

1. MakeSet(x) — утворює нову підмножимну для елемента x
2. Union(r, s) — об'єднує підмножини з представниками r та s і назначає r новим представником
3. Find(x) — визначає підмножину, до якої належить елемент x та повертає представника підмножини

Використання

- Блокування надлишкових зв'язків в Ethernet-мережах
- Сегментування зображень
- Генерація лабіринтів

Реалізація

```cpp
#include<iostream>
#include<algorithm>

using namespace std;

class DSU {
public:
    void MakeSet(int x) {
        p[x] = x;
        rank[x] = 0;
    }

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

    void Unite(int x, int y) {
        x = Find(x);
        y = Find(y);

        if (x == y) return;
        else if (rank[x] < rank[y]) p[x] = y;
        else {
            p[y] = x;
            if (rank[x] == rank[y]) ++rank[x];
        }
    }

private:

    int p[1000], rank [1000];
};

```

Реалізація

Покращення:

- Стиснення шляхів (path compression)

```cpp
#include<iostream>
#include<algorithm>

using namespace std;

class DSU {
public:
    void MakeSet(int x) {
        p[x] = x;
        rank[x] = 0;
    }

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

    void Unite(int x, int y) {
        x = Find(x);
        y = Find(y);

        if (x == y) return;
        else if (rank[x] < rank[y]) p[x] = y;
        else {
            p[y] = x;
            if (rank[x] == rank[y]) ++rank[x];
        }
    }

private:

    int p[1000], rank [1000];
};

```

Реалізація

Покращення:

- Стиснення шляхів (path compression)
- Random-Union

```cpp
#include<iostream>
#include<algorithm>

using namespace std;

class DSU {
public:
    void MakeSet(int x) {
        p[x] = x;
        rank[x] = 0;
    }

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

    void Unite(int x, int y) {
        x = Find(x);
        y = Find(y);
        if (rand() % 2 == 0) {
            swap(x, y);
        }
    }

private:

    int p[1000], rank [1000];
};

```

Дякую за увагу