CHM (наивные реализации)

Система (лес, объединение) непересекающихся множеств (CHM, disjoint set forest, DSF, disjoint set union, DSU) — иерархическая структура данных, позволяющая эффективно работать с множествами.

Содержание

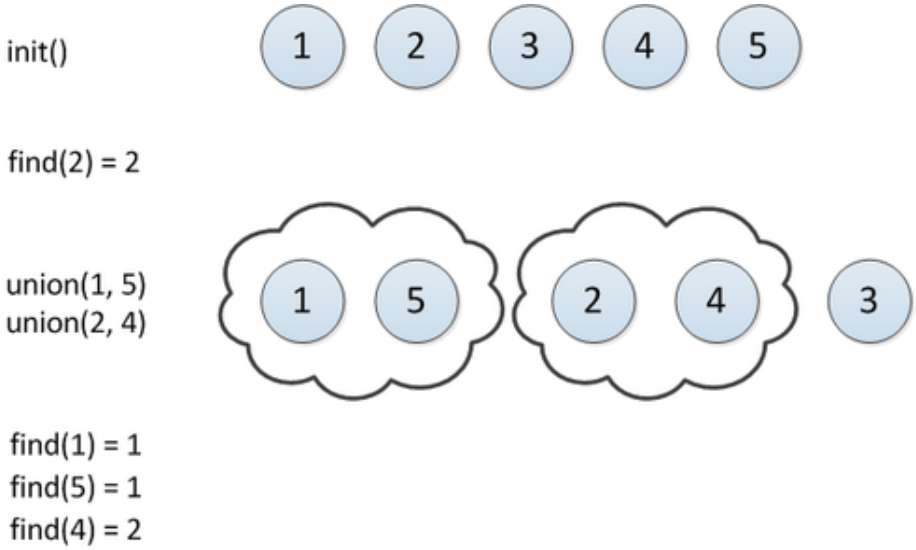
Описание

Структура хранит набор объектов (например, чисел от 0 до $n - 1$) в виде непересекающихся множеств. У каждого множества есть конкретный представитель.

Определены две операции:

- $\mathrm{union(x, y)}$ — объединяет множества, содержащие x и y
- $\mathrm{find(x)}$ — возвращает представителя множества, в котором находится x

Для любого элемента множества представитель всегда одинаковый. Поэтому чтобы проверить принадлежность элементов x и y одному множеству достаточно сравнить $\mathrm{find(x)}$ и $\mathrm{find(y)}$.



Реализации

С помощью массива

Пусть в массиве s хранятся номера множеств, в $s[i]$ будет храниться номер множества, к которому принадлежит i. Этот номер отождествляет множество, find возвращает именно его. Тогда find, очевидно, будет работать за $O(1)$.

Чтобы объединить множества x и y, надо изменить все $s[i]$, равные номеру множества x, на номер y. Тогда union работает за $O(n)$.

```
int s[n]
func init():
    for i = 0 to n - 1
        s[i] = i                // сначала каждый элемент лежит в своем множестве
```

```
int find(k):
    return s[k]
```

```
func union(x, y):
    if s[x] == s[y]
        return
    else
        t = s[y]
        for i = 0 to n - 1
            if s[i] == t
                s[i] = s[x]
```

С помощью списка

Будем хранить множество в виде списка. Для каждого элемента списка храним ссылку на следующий элемент и указатель на $head$, который является представителем. Для того чтобы найти представителя, нужно перейти по ссылке на $head$. Значит find работает за $O(1)$.

Для объединения множеств потребуется объединить два списка и обновить ссылки на $head$. Таким образом, union работает за $O(n)$. Чтобы объединить два списка, нужно хранить ссылку на $tail$. Ее можно хранить в голове списка.

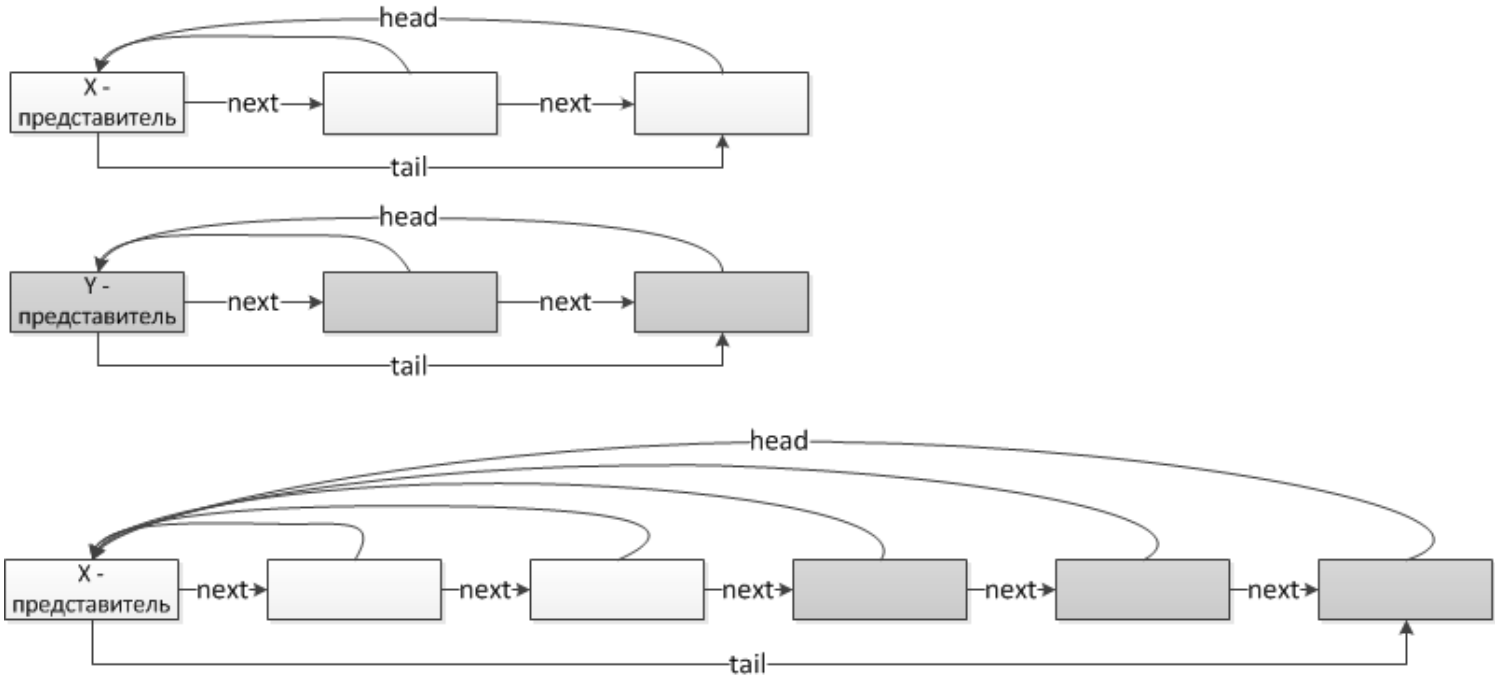
```
struct SetItem
    int data
    SetItem head
    SetItem next
    SetItem tail
```

```
SetItem s[n]
```

```
func init():
    for i = 0 to n - 1
        s[i].data = i
        s[i].head = s[i]
        s[i].tail = s[i]
        s[i].next = null
```

```
int find(SetItem x):                    // подразумевается, что x — ссылка на один из элементов
    return x.head.data
```

```
func union(SetItem x, SetItem y):  // x и y — элементы множеств
    x = x.head
    y = y.head
    if x == y
        return
    x.tail.next = y                     // соединим списки
    x.tail = y.tail                     // сделаем корректную ссылку на tail в head
    while y ≠ null                      // скорректируем ссылки на head у элементов множества y
        y.head = x
        y = y.next
```



Другие реализации

- CHM (списки с весовой эвристикой)
- CHM (реализация с помощью леса корневых деревьев)

Источники информации

- Википедия — Система непересекающихся множеств (http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BD%D0%B5%D0%BF%D0%B5%D1%80%D0%B5%D1%81%D0%B5%D0%BA%D0%B0%D1%8E%D1%89%D0%B8%D1%85%D1%81%D1%8F_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2)
- Система непересекающихся множеств и её применения (http://habrahabr.ru/blogs/algorithm/104772/)
- Т. Кормен - Алгоритмы, построение и анализ. Второе издание. Часть V. Глава 21.

Источник — «http://neerc.ifmo.ru/wiki/index.php?title=CHM_(наивные_реализации)&oldid=66257»

- Эта страница последний раз была отредактирована 27 сентября 2018 в 11:54.