Union-Find

```
class UnionFind(object):
    def __init__(self, N):
        self.group = N                  # all disjoint
        self.parent = list(range(N))    # point to self
        self.rank = [0] * N             # approx subtree height

    def find(self, p):
        parent = self.parent
        while p != parent[p]:
            parent[p] = parent[parent[p]] # path compression
            p = parent[p]
        return p

    def union(self, p, q):
        if p == q: return
        i, j = self.find(p), self.find(q)
        if i == j: return
        self.group -= 1
        parent, rank = self.parent, self.rank
        if rank[i] > rank[j]:
            parent[j] = i
        elif rank[i] < rank[j]:
            parent[i] = j
        else:
            parent[i] = j
            rank[j] += 1

class UnionFind_Hash(object):
    def __init__(self):
        self.group = 0
        self.parent = {}
        self.rank = {}

    def add(self, p):
        if p not in self.parent:
            self.group += 1
            self.parent[p] = p
            self.rank[p] = 0
```

union-find on graph

Number of Friend Circles `count group`
Number of Connected Components in an Undirected Graph `count group`
Graph Valid Tree `并查集检查 no cycle` `one group`
Redundant Connection I (undirected) `并查集检查 no cycle`
Redundant Connection II (directed) **`两个父节点? 删哪一个?`**

union-find on 2d grid

Surrounded Regions `imaginary boarder node`
Max Area of Island `union find`
Max Area of Island (add 1) (Making A Large Island) `union find` `group_size`
Number of Islands I `count group`
Number of Islands II (addLand) **`hash union-find`**

union-find on hash (item2id)

Sentence Similarity II (transitive) `words, word2id`
Accounts Merge `emails, email2id`
Couples Holding Hands `person2seat`
Longest Consecutive Sequence **`val2ind, unionfind, ! duplicate`**
Similar String Groups `piecewise`

smart

Couples Holding Hands `person2seat`
Most Stones Removed with Same Row or Column `Connected stones can be reduced to 1 stone` `stones number - islands number` **`unionfind on 行/列`**
Longest Consecutive Sequence **`val2ind, unionfind, ! duplicate`**
Bricks Falling When Hit **`reverse add brick`** **`uf.sz[uf.find(-1)]`** **`cannot be negative`**