algos-py Documentation

Release 0.4.4

Aleksandr Lisianoi

Dec 06, 2021

Contents:

1 Disjoint Set Union 3

2 Two-dimensional lines 5

3 Indices and tables 7

Index 9

Warning: This is very much a work in progress.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

CHAPTER 1

Disjoint Set Union

class src.dsu.**DisjointSetUnion**(*xs=None*)

Construct a disjoint set union (DSU)

This data structure tracks a set of elements partitioned into a number of *disjoint* (i.e. non-overlapping) subsets. It provides almost constant-time operations to:

1. Add a new element to the set
2. Join two sets into one (union)
3. Find if two elements are in the same set (find)

Alternative names for this data structure:

1. Disjoint-set data structure
2. Union-find data structure
3. Merge-find set

__init__(*xs=None*)

Create an empty or a pre-populated disjoint set union (DSU)

Parameters **xs** – a list of elements. If None, then an empty DSU is created. Otherwise, each element is treated as its own set.

find(*key*, *w=0*)

Find the set of the provided key

This method returns the same key for those keys that belong to the same set. Internally, the keys from the same set make up a tree whose root holds the key that represents the whole set.

When you call find with a specific key, there are two edge cases:

1. The provided key is, in fact, the root of the tree. If so, it is immediately returned. 2. The provided key is a leaf of some very long path to the root of the tree. In this case, the entire path to the root must be traversed.

To avoid having very long paths, the second case also performs path compression after traversing the path to the root.

Parameters

- **key** – the key whose set should be found
- **w** – the weight of the subtree of the key

union(*key1*, *key2*)

Combine the sets that contain the two given keys into one

If both keys belong to the same set, the method does not achieve much. However, path compression could still change the underlying data structure.

Parameters

- **key1** – the key contained in the first set
- **key2** – the key contained in the second set

CHAPTER 2

Two-dimensional lines

`src.geometry.line2d.`**`standard_line`**(*x0*, *y0*, *x1*, *y1*)

Computes the coefficients of Ax + By + C = 0 given two plane points.

Returns a tuple (A, B, C); it is (0, 0, 0) if both points are the same

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Index

Symbols

__init__() (*src.dsu.DisjointSetUnion method*), 3

D

DisjointSetUnion (*class in src.dsu*), 3

F

find() (*src.dsu.DisjointSetUnion method*), 3

S

standard_line() (*in module src.geometry.line2d*), 5

U

union() (*src.dsu.DisjointSetUnion method*), 4