Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
فارسی
Polski
ไทย
Edit links

Tarjan's off-line lowest common ancestors algorithm

From Wikipedia, the free encyclopedia
(Redirected from Tarjan's off-line least common ancestors algorithm)

Not to be confused with Tarjan's strongly connected components algorithm.

In computer science, **Tarjan's off-line lowest common ancestors algorithm** is an algorithm for computing lowest common ancestors for pairs of nodes in a tree, based on the union-find data structure. The lowest common ancestor of two nodes *d* and *e* in a rooted tree *T* is the node *g* that is an ancestor of both *d* and *e* and that has the greatest depth in *T*. It is named after Robert Tarjan, who discovered the technique in 1979. Tarjan's algorithm is *offline*; that is, unlike other lowest common ancestor algorithms, it requires that all pairs of nodes for which the lowest common ancestor is desired must be specified in advance. The simplest version of the algorithm uses the union-find data structure, which unlike other lowest common ancestor data structures can take more than constant time per operation when the number of pairs of nodes is similar in magnitude to the number of nodes. A later refinement by Gabow & Tarjan (1983) speeds the algorithm up to linear time.

Pseudocode [edit]

The pseudocode below determines the lowest common ancestor of each pair in *P*, given the root *r* of a tree in which the children of node *n* are in the set *n.children*. For this offline algorithm, the set *P* must be specified in advance. It uses the *MakeSet*, *Find*, and *Union* functions of a disjoint-set forest. *MakeSet(u)* removes *u* to a singleton set, *Find(u)* returns the standard representative of the set containing *u*, and *Union(u,v)* merges the set containing *u* with the set containing *v*. TarjanOLCA(*r*) is first called on the root *r*.

```
function TarjanOLCA(u)
    MakeSet(u);
    u.ancestor := u;
    for each v in u.children do
        TarjanOLCA(v);
        Union(u,v);
        Find(u).ancestor := u;
    u.colour := black;
    for each v such that {u,v} in P do
        if v.colour == black
            print "Tarjan's Lowest Common Ancestor of " + u +
                  " and " + v + " is " + Find(v).ancestor + ".";
```

Each node is initially white, and is colored black after it and all its children have been visited. The lowest common ancestor of the pair *{u,v}* is available as *Find(v).ancestor* immediately (and only immediately) after *u* is colored black, provided *v* is already black. Otherwise, it will be available later as *Find(u).ancestor*, immediately after *v* is colored black.

For reference, here are optimized versions of *MakeSet*, *Find*, and *Union* for a disjoint-set forest:

```
function MakeSet(x)
    x.parent := x
    x.rank   := 0

function Union(x, y)
    xRoot := Find(x)
    yRoot := Find(y)
    if xRoot.rank > yRoot.rank
        yRoot.parent := xRoot
    else if xRoot.rank < yRoot.rank
        xRoot.parent := yRoot
    else if xRoot != yRoot
        yRoot.parent := xRoot
        xRoot.rank := xRoot.rank + 1

function Find(x)
```

```
    if x.parent == x
       return x
    else
       x.parent := Find(x.parent)
       return x.parent
```

References [edit]

- Gabow, H. N.; Tarjan, R. E. (1983), "A linear-time algorithm for a special case of disjoint set union", *Proceedings of the 15th ACM Symposium on Theory of Computing (STOC)*, pp. 246–251, doi:10.1145/800061.808753.
- Tarjan, R. E. (1979), "Applications of path compression on balanced trees", *Journal of the ACM* **26** (4): 690–715, doi:10.1145/322154.322161.

Categories: Graph algorithms