Algorithm Assignment 7

> *10185101210 陈俊潼*

24.1-4

Add a property `marked` for each vertices. The modified version of Bellman-Ford algorithm is as follows:

```
MODIFIED_BELLMAN_FORD(G, w, s):
    INITIALIZE-SINGLE-SOURCE(G, s)
    for i = 1 to |G.v| - 1
        for each edge(u,v) in G.E:
            RELAX(u, v, w)
    for each edge(u, v) in G.E:
        if v.d > u.d + w(u, v):
            v.marked = true
    for each vertex where v.marked = true:
        MARK_LOOP(v)
```

And then define `CHECK_LOOP` as a function to determine if there exists a loop with a negative-weight cycle:

```
MARK_LOOP(v):
    if v != NULL and v.d != -inf
        v.d = -inf
        MARK_LOOP(v)
    return
```

In contrast with the previous version, this algorithm do not only mark v that is the destination with a negative-weight loop, but also mark every vertex that's its prudence vertex.

24.3-4

For each edge (u, v) of the generated shortest-path tree, check if the two condition holds:

1. For each edge (u, v), $v.d \leq u.d + w(u, v)$ holds.
2. For each vertex v, $v.d = v.\pi.d + w(v.\pi, v)$ holds.

If either condition is tested to be false, then the generated result is incorrect.

Since this algorithm needs to traverse all vertices and edge once and only once, the complexity of this algorithm is $O(V + E)$.