Open Topic 负权重环

丁保荣 171860509

问题描述：请证明BF算法一定能够找到负权重环，请改造BF算法，输出负权重环

- 思路：
- 1. 证明BF算法一定能够找到负权重环(如果存在的话)
- 2. 改造BF算法，输出负权重环
- 3. 证明改造后的BF算法的正确性

1.证明BF算法一定能够找到负权重环(如果存在的话)

```
BELLMAN-FORD(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
for i = 1 to |G.V| - 1
    for each edge (u, v) ∈ G.E
        RELAX(u, v, w)
for each edge (u, v) ∈ G.E
    if v.d > u.d + w(u, v)
        return FALSE
return TRUE
```

当无负权重环的时候，Bellman-Ford算法输出True

当有负权重环的时候，Bellman-Ford算法输出False

我们要证明的是Bell-Ford算法对于有无负权重环的判断是正确的

即证明：Bellman-Ford算法输出False当且仅当G有负权重环

1.证明BF算法一定能够找到负权重环(如果存在的话)

即证明：Bellman-Ford算法输出False当且仅当G有负权重环

等价于证明以下两个命题

命题 1.1 如果BF算法输出False，则G有负权重环

命题 1.2 如果G有负权重环，则BF算法输出False

通过逆否命题的性质，我们可以对定理 1.1做逆否得到命题1.3

命题 1.3 如果G没有负权重环，则BF算法输出True

命题1.1与命题1.3等价，所以我们只要证明命题1.2和命题1.3就可以了

```
BELLMAN-FORD(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
for i = 1 to |G.V| - 1
    for each edge (u, v) ∈ G.E
        RELAX(u, v, w)
for each edge (u, v) ∈ G.E
    if v.d > u.d + w(u, v)
        return FALSE
return TRUE
```

1.证明BF算法一定能够找到负权重环(如果存在的话)

命题 1.2 如果G有负权重环，则BF算法输出False

证明

我们用反证法说明：假设BF算法输出True

设负权重环 l 为 $(v_0, v_1, ..., v_k, v_0)$, $w(l) = w(v_0, v_1) + w(v_1, v_2) + ... + w(v_k, v_0) < 0$. 因为 BF 算法输出为 True，对于算法第六行的判断，所有的边都不成立，即任意边 (u, v) 满足 $v.d \le u.d + w(u, v)$,

所以 $v_1.d \le v_0.d + w(v_0, v_1)$,..., $v_k.d \le v_{k-1}.d + w(v_{k-1}, v_k)$, $v_0.d \le v_k.d + w(v_k.d, v_0.d)$.

对这些进行累加，我们得到: $v_1.d + v_2.d + ... + v_k.d + v_0.d \le v_0.d + v_1.d + ... + v_k.d + w(v_0, v_1) + w(v_1, v_2) + ... + w(v_k, v_0)$

所以有:$0 \le w(l)$ 与 $w(l) < 0$ 矛盾，所以假设不成立。所以 BF 算法输出 False。

```
BELLMAN-FORD(G, w, s)
1  INITIALIZE-SINGLE-SOURCE(G, s)
2  for i = 1 to |G.V| - 1
3      for each edge (u, v) ∈ G.E
4          RELAX(u, v, w)
5  for each edge (u, v) ∈ G.E
6      if v.d > u.d + w(u, v)
7          return FALSE
8  return TRUE
```

1.证明BF算法一定能够找到负权重环(如果存在的话)

命题 1.3 如果G没有负权重环，则BF算法输出True

我们这里引入算导上给出的几个性质：路径松弛性质、三角不等式性质、收敛性质

因为不含负权重环，(也不会含正权重环，书上有证明)，设 s 到任意可通达节点 v 的最短路径经过的节点数一定小于等于 $|V|$(包括 s 和 v), 所以最短路经过的边数一定小于等于 $|V|-1$.

因为 BF 算法中有 $|V-1|$ 轮的迭代，所以这条最短路径上的边，一定可以按照相应的顺序被松弛。所以根据路径松弛性质，我们可以所有的可通达节点 v 有 $v.d=\delta(s,v)$.

然后根据收敛性质, 在之后的迭代中总有 $v.d=\delta(s,v)$.

最后根据三角不等式性质，有 $v.d=\delta(s,v)\le\delta(s,u)+w(u,v)=u.d+w(u,v)$.

所以所有可通达节点都不满足第六行的判断，所以最后输出 True.

```
BELLMAN-FORD(G, w, s)
1  INITIALIZE-SINGLE-SOURCE(G, s)
2  for i = 1 to |G.V| - 1
3      for each edge (u, v) ∈ G.E
4          RELAX(u, v, w)
5  for each edge (u, v) ∈ G.E
6      if v.d > u.d + w(u, v)
7          return FALSE
8  return TRUE
```

1.证明BF算法一定能够找到负权重环(如果存在的话)

```
BELLMAN-FORD(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
for i = 1 to |G.V| - 1
    for each edge (u, v) ∈ G.E
        RELAX(u, v, w)
for each edge (u, v) ∈ G.E
    if v.d > u.d + w(u, v)
        return FALSE
return TRUE
```

至此，BF算法一定能够找到负权重环已经证明(如果存在的话)

下面证明改进BF算法使其能输出负权重环

2.改造BF算法，输出负权重环

Algorithm 1 BELLMAN-FORD(G, w, s)

解答:

```
INITIALIZE-SINGLE-SOURCE(G, s)
for i = 1 to |G.V| − 1 do
    for each edge(u, v) ∈ G.E do
        RELAX(u, v, w)
    end for
end for
for each u ∈ G.V do
    MAKE-SET(u)
end for
for each edge(u, v) ∈ G.E do
    if v.d > u.d + w(u, v) then
        v.d = u.d + w(u, v)
        v.π = u
        t = v
        while t.π! = NIL do
            if !SAME-SET(v, t) then
                UNION-SET(v, t)
            else
                p = t
                repeat
                    Print p
```

```
        MAKE-SET(u)
end for
for each edge(u, v) ∈ G.E do
    if v.d > u.d + w(u, v) then
        v.d = u.d + w(u, v)
        v.π = u
        t = v
        while t.π != NIL do
            if !SAME-SET(v, t) then
                UNION-SET(v, t)
            else
                p = t
                repeat
                    Print p
                    p = p.π
                until p != t
            end if
        end while
        return FALSE
    end if
end for
return TRUE
```

2.改造BF算法，输出负权重环

Algorithm 1 BELLMAN-FORD(G, w, s)

解答:

```
INITIALIZE-SINGLE-SOURCE(G, s)
for i = 1 to |G.V| − 1 do
    for each edge(u, v) ∈ G.E do
        RELAX(u, v, w)
    end for
end for
for each u ∈ G.V do
    MAKE-SET(u)
end for
for each edge(u, v) ∈ G.E do
    if v.d > u.d + w(u, v) then
        v.d = u.d + w(u, v)
        v.π = u
        t = v
        while t.π! = NIL do
            if !SAME-SET(v, t) then
                UNION-SET(v, t)
            else
                p = t
                repeat
                    Print p
                    p = p.π
                until p! = t
            end if
        end while
        return FALSE
    end if
end for
return TRUE
```

核心思想：1. 进行BF算法

2.多进行一轮松弛操作，如果该节点被更新了，那么我们从这个节点开始不断找前驱，直到重复的节点，那么就会出现一个环

举个例子

3. 证明改造后的BF算法的正确性

我们主要证明两点

定理3.1.BF算法生成的前驱图如果有环，那么这个环一定是负环。

定理3.2.如果在第|V|次迭代中，我们仍能更新节点的值，那么我们一定能从这个节点出发找到一个环

3. 证明改造后的BF算法的正确性

定理3.1.BF算法生成的前驱图如果有环，那么这个环一定是负环。

假设 $G.\pi$ 中存在非负环 $c = \{v_0, v_1, ..., v_k, v_0\}$, 且最后被松弛形成环的边是 (v_k, v_0).
那么这条边被松弛的时候有 $v_0 > v_k + w(v_k, v_0) = (v_{k-1} + w(v_{k-1}, v_k)) + w(v_k, v_0) = ... = v_0 + w(c)$.
所以我们得到 $w(c) < 0$, 与非负环 c 矛盾，所以假设不成立。所以 $G.\pi$ 不存在非负环。
所以 $G.\pi$ 如果有环，一定是负环。

3. 证明改造后的BF算法的正确性

定理3.2 .如果在第|V|次迭代中，我们仍能更新节点的值，那么我们一定能从这个节点出发找到一个环

我们先证明两个引理：

引理3.3: 如果s到v的最短路径是简单路径(即不含环)，那么v.d在经过|V|-1轮的迭代后就已经确定

根据路径松弛性质，易证

引理3.4：所有可通达节点除了s都有前驱。若s有前驱，则s.d<0

这个易证，因为v在更新v.d的时候也指定了v.pi

如果s有前驱，说明进行了松弛，那么新的s.d肯定小于原来的s.d，而原来的s.d=0. 所以若s有前驱，则s.d<0.

3. 证明改造后的BF算法的正确性

定理3.2 .如果在第|V|次迭代中，我们仍能更新节点的值，那么我们一定能从这个节点出发找到一个环

引理3.3: 如果s到v的最短路径是简单路径(即不含环)，那么v.d在经过|V|-1轮的迭代后就已经确定

引理3.4：所有可通达节点除了s都有前驱。若s有前驱，则s.d<0

我们现在证明定理3.2

假设我们在第|V|次迭代中，更新了v.d的值，但我们从v出发没有找到环。

那么根据引理3.4，说明最后停留在s处且s.d=0。此时说明s到v的最短路径就是简单路径，但根据引理3.3，s到v的简单路径在|V|-1轮迭代中就更新完成，不会再|V|轮迭代的时候更新，所以与题意矛盾，所以假设不成立

谢谢大家！