FIND-AUGMENTING-PATH($G_{M,h}$)

```
1   Q = ∅
2   F_L = ∅
3   F_R = ∅
4   for each unmatched vertex l ∈ L
5       l.π = NIL
6       ENQUEUE(Q, l)
7       F_L = F_L ∪ {l}        // forest F starts with unmatched vertices in L
8   repeat
9       if Q is empty          // ran out of vertices to search from?
10          δ = min {l.h + r.h − w(l, r) : l ∈ F_L and r ∈ R − F_R}
11          for each vertex l ∈ F_L
12              l.h = l.h − δ       // relabel according to equation (25.5)
13          for each vertex r ∈ F_R
14              r.h = r.h + δ       // relabel according to equation (25.5)
15          from G, M, and h, form a new directed equality graph G_{M,h}
16          for each new edge (l, r) in G_{M,h}     // continue search with new edges
17              if r ∉ F_R
18                  r.π = l                         // discover r, add it to F
19                  if r is unmatched
20                      an M-augmenting path has been found
                            (exit the repeat loop)
21                  else ENQUEUE(Q, r)              // can search from r later
22                      F_R = F_R ∪ {r}
23      u = DEQUEUE(Q)                              // search from u
24      for each neighbor v of u in G_{M,h}
25          if v ∈ L
26              v.π = u
27              F_L = F_L ∪ {v}                     // discover v, add it to F
28              ENQUEUE(Q, v)                       // can search from v later
29          elseif v ∉ F_R                          // v ∈ R, do same as lines 18–22
30              v.π = u
31              if v is unmatched
32                  an M-augmenting path has been found
                        (exit the repeat loop)
33              else ENQUEUE(Q, v)
34                  F_R = F_R ∪ {v}
35  until an M-augmenting path has been found
36  using the predecessor attributes π, construct an M-augmenting path P
        by tracing back from the unmatched vertex in R
37  return P
```