

```

a 0 0: c d parent: 0
b 0 0: d f parent: 0
c 0 0: a e parent: 0
d 0 0: a b e f parent: 0
e 0 0: c d parent: 0
f 0 0: b d g h parent: 0
g 0 0: f parent: 0
h 0 0: f parent: 0
DFS(a)
  num(a) = pred(a) = 1
  trying edge(ac)
  push(edge(ac))
  DFS(c)
    num(c) = pred(c) = 2
    trying edge(ca)
    trying edge(ce)
    push(edge(ce))
    DFS(e)
      num(e) = pred(e) = 3
      trying edge(ec)
      trying edge(ed)
      push(edge(ed))
      DFS(d)
        num(d) = pred(d) = 4
        trying edge(da)
        push(edge(da))
        pred2(d) = 1 (= num(a))
        trying edge(db)
        push(edge(db))
        DFS(b)
          num(b) = pred(b) = 5
          trying edge(bd)
          trying edge(bf)
          push(edge(bf))
          DFS(f)
            num(f) = pred(f) = 6
            trying edge(fb)
            trying edge(fd)
            push(edge(fd))
            pred2(f) = 4 (= num(d))
            trying edge(fg)
            push(edge(fg))
            DFS(g)
              num(g) = pred(g) = 7
              trying edge(gf)
              BLOCK: edge(fg)
              trying edge(fh)
              push(edge(fh))
              DFS(h)
                num(h) = pred(h) = 8
                trying edge(hf)
                BLOCK: edge(fh)
                pred(b) = 4 (= pred(f))
                BLOCK: edge(fd) edge(bf) edge(db)
                trying edge(de)
                trying edge(df)
                pred(e) = 1 (= pred(d))
                pred(c) = 1 (= pred(e))
                BLOCK: edge(da) edge(ed) edge(ce) edge(ac)
                trying edge(ad)
a 1 1: c d parent: 0
b 5 4: d f parent: d
c 2 1: a e parent: a
d 4 1: a b e f parent: e
e 3 1: c d parent: c
f 6 4: b d g h parent: b
g 7 7: f parent: f
h 8 8: f parent: f

```

output

class ← inside it an array of linked list  
 class ← edge

initialize the graph in a constructor (nodes and neighbors)

indicate a neighbour by an index

use a linked list of a linked list  
 or an array of linked list

global variable

at the beginning of recursion going  
 increment at the beginning  
 decrement before you exit

loop  
 setw

pg. 470