

not in the exam

2/20/2014

Program

Connectivity in undirected graphs

```

blockDFS(v)
  pred(v) = num(v) = i++;
  for all vertices u adjacent to v
    if edge(vu) is not on stack
      push(edge(vu));
    if num(u) is 0
      blockDFS(u);
    if pred(u) >= num(v)
      e = pop();
      while e != edge(vu)
        output e;
        e = pop();
      output e;
      // e == edge(vu);
      pred(v) = min(pred(v), pred(u)); // take a predecessor higher up in the tree;
    else if u is not the parent of v
      pred(v) = min(pred(v), num(u)); // update when back edge(vu) is found;

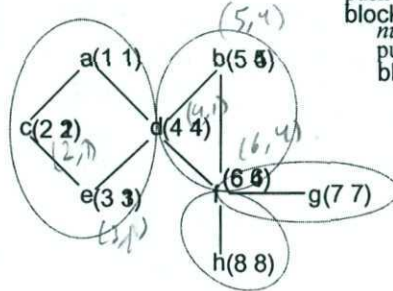
blockSearch()
  for all vertices v
    num(v) = 0;
  i = 1;
  while there is a vertex v such that num(v) == 0
    blockDFS(v);

```

when you finish from block stack hit

u -> after

Connectivity in undirected graphs: example



- edge(fb)
 - edge(fd)
 - edge(bf)
 - edge(db)
 - edge(da)
 - edge(ed)
 - edge(ce)
 - edge(ac)
- stack

```

blockDFS(a)
  num(a) = pred(a) = 1
  push edge(ac)
  blockDFS(c)
    num(c) = pred(c) = 2
    push edge(ce)
    blockDFS(e)
      num(e) = pred(e) = 3
      push edge(ed)
      blockDFS(d)
        num(d) = pred(d) = 4
        push edge(da)
        pred(d) = min(pred(d), num(a)) = 1
        push edge(db)
        blockDFS(b)
          num(b) = pred(b) = 5
          push edge(bf)
          blockDFS(f)
            num(f) = pred(f) = 6
            push edge(fd)
            pred(f) = min(pred(f), num(d)) = 4
            push edge(fg)
            blockDFS(g)
              num(g) = pred(g) = 7
              output edge(fg)
              push edge(fh)
              blockDFS(h)
                num(h) = pred(h) = 8
                output edge(fh)
            pred(b) = min(pred(b), pred(f)) = 4
            output edge(fd), edge(bf), edge(db)
          pred(e) = min(pred(e), pred(d)) = 1
          pred(c) = min(pred(c), pred(e)) = 1
          output edge(da), edge(ed), edge(ce), edge(ac)

```