§1 EULER-TRAIL INTRO 1

1. Intro. We output an Eulerian trail of the (undirected) graph named on the command line. (Each edge is considered to be two directed arcs; thus it is traversed in both directions.)

If the graph isn't connected, we consider only the vertices that are reachable from the first one, g-vertices.

```
#include <stdio.h>
#include <stdlib.h>
#include "gb_graph.h"
#include "gb_save.h"
  (Subroutine 3)
  main(\mathbf{int} \ argc, \mathbf{char} * argv[])
     register int k;
     Graph *g;
     Vertex *u, *v;
     \mathbf{Arc} *a;
     \langle \text{Input the graph } 2 \rangle;
     \langle \text{Traverse depth first 4} \rangle;
     \langle \text{Output the trail 5} \rangle;
     printf("\n");
  }
2. \langle \text{Input the graph 2} \rangle \equiv
  if (argc \neq 2 \lor \neg(g = restore\_graph(argv[1]))) {
     fprintf(stderr, "Usage: \_\%s\_foo.gb\n", argv[0]);
     exit(-1);
  fprintf(stderr, "OK, LI've_Linput_L''s', \n", argv[1]);
  gb\_new\_edge(g\neg vertices, g\neg vertices + g\neg n, 0); /* dummy edge */
This code is used in section 1.
```

**3.** Subroutine dfs(u, v) sets  $v \neg parent = u$  and  $v \neg nav$  to the vertex that follows u in v's adjacency list. It also explores all vertices reachable from v that haven't already been seen.

```
#define parent v.V
#define nav \ w.A
\langle \text{Subroutine 3} \rangle \equiv 
void dfs(\text{register Vertex }*u,\text{register Vertex }*v)
\{ \\ \text{register Vertex }*w; \\ \text{register Arc }*a; \\ v \text{-} parent = u; \\ \text{for } (a = v \text{-} arcs; \ a; \ a = a \text{-} next) \ \{ \\ w = a \text{-} tip; \\ \text{if } (w \equiv u) \ v \text{-} nav = a \text{-} next; \\ \text{else if } (w \text{-} parent \equiv \Lambda) \ dfs(v,w); \\ \} \\ \}
This code is used in section 1.
```

4.  $\langle \text{Traverse depth first } 4 \rangle \equiv dfs(g \rightarrow vertices + g \rightarrow n, g \rightarrow vertices);$ This code is used in section 1. 2 INTRO EULER-TRAIL §5

 ${\bf 5.}$   $\,$  Now the Eulerian traversal is beautifully simple.

This code is used in section 1.

 $\S 6$  Euler-trail index 3

## 6. Index.

```
a: \underline{1}, \underline{3}.
Arc: 1, 3.
arcs: 3, 5.
\begin{array}{ccc} argc \colon & \underline{1}, & 2. \\ argv \colon & \underline{1}, & 2. \end{array}
dfs: \underline{3}, 4.
exit: 2.
fprintf: 2.
g: \underline{1}.
gb\_new\_edge: 2.
Graph: 1.
k: \underline{1}.
main: \underline{1}.
name: 5.
nav: \underline{3}, 5.
next: 3, 5.
parent: \underline{3}.
printf: 1, 5.
restore\_graph: 2.
stderr: 2.
tip: 3, 5.
u: \ \ \underline{1}, \ \underline{3}. \ v: \ \underline{1}, \ \underline{3}.
Vertex: 1, 3.
vertices: 1, 2, 4, 5.
w: \underline{3}.
```

## 4 NAMES OF THE SECTIONS

EULER-TRAIL

```
 \begin{array}{lll} \left\langle \text{Input the graph 2} \right\rangle & \text{Used in section 1.} \\ \left\langle \text{Output the trail 5} \right\rangle & \text{Used in section 1.} \\ \left\langle \text{Subroutine 3} \right\rangle & \text{Used in section 1.} \\ \left\langle \text{Traverse depth first 4} \right\rangle & \text{Used in section 1.} \\ \end{array}
```

## EULER-TRAIL

	Section	Page
Intro		1
Indev	6	•