§1 SAND INTRO 1

(Downloaded from https://cs.stanford.edu/~knuth/programs.html and typeset on September 17, 2017)

1. Intro. This program was written (somewhat hastily) in order to experiment with sandpiles.

The first command line argument is the name of a file that specifies an undirected graph in Stanford GraphBase SAVE_GRAPH format; the graph may have repeated edges, but it must not contain loops. It should be connected. It shouldn't have more than 100 vertices. I don't check these assumptions.

An optional second argument is the number of the root vertex.

```
#include "gb_graph.h"
#include "gb_save.h"
          (Preprocessor definitions)
         int vec[1000][1000];
         int x[1000], d[1000], t[1000];
         int n, r;
          ⟨Subroutines 4⟩
          main(\mathbf{int} \ argc, \mathbf{char} * argv[])
                  register int j, k;
                    Vertex * v;
                   Arc * a;
                   Graph * g;
                   \langle \text{Input the graph } 2 \rangle;
                    \langle \text{ Prepare the } vec \text{ table } 3 \rangle;
                    \langle \text{ Reduce the vector } d \ 5 \rangle;
          }
2. \langle \text{Input the graph 2} \rangle \equiv
        if (argc < 2) {
                  fprintf(stderr, "Usage: \_\%s\_foo.gb_\_[r] \n", argv[0]);
                   exit(1);
         g = restore\_graph(argv[1]);
         if (\neg g) {
                  fprintf(stderr, "Sorry, \_can't\_create\_the\_graph\_from\_file\_%s!\_(error\_code\_%d)\n", argv[1], 
                                      panic\_code);
                   exit(-1);
          }
         n = g \rightarrow n;
         if (argc > 2) sscanf(argv[2], "%d", &r);
This code is used in section 1.
```

2 INTRO SAND §3

```
3. \langle \text{Prepare the } vec \text{ table } 3 \rangle \equiv
  for (j = 0; j < n; j \leftrightarrow) {
     v = g \rightarrow vertices + j;
      for (a = v \rightarrow arcs; a; a = a \rightarrow next) {
         k = a \rightarrow tip - g \rightarrow vertices;
         d[j]++;
         vec[j][k]--;
      vec[j][j] = d[j];
  if (r) {
       \mathbf{for} \ (j = 0; \ j < n; \ j +\!\!\!\!+) \ \ k = vec[0][j], vec[0][j] = vec[r][j], vec[r][j] = k; 
       \mathbf{for} \ (j = 0; \ j < n; \ j + +) \ k = vec[j][0], vec[j][0] = vec[j][r], vec[j][r] = k; 
      k = d[0], d[0] = d[r], d[r] = k;
This code is used in section 1.
4. The reduce subroutine topples a given vector x until it is stable.
\langle \text{Subroutines 4} \rangle \equiv
   void reduce()
   {
      register int j, k, h;
      while (1) {
         h = 0;
        for (j = 1; j < n; j++)
           if (x[j] \ge d[j]) {
              h=1;
              for (k = 1; k < n; k++) x[k] -= vec[j][k];
         if (h \equiv 0) break;
This code is used in section 1.
```

 $\S5$ SAND INTRO 3

```
5. \langle Reduce the vector d 5\rangle \equiv
  printf("The duvector is");
  for (j = 1; j < n; j ++) {
     x[j] = d[j];
    printf(" \sqcup %d", x[j]);
  printf("\n_and_it_reduces_ito");
  reduce();
  for (j = 1; j < n; j ++) {
    printf(" " " " ", x[j]);
    x[j] = d[j] - x[j];
  printf("\nThe_
ut_
uvector_
uis");
  reduce();
  for (j = 1; j < n; j ++) {
    printf(" \sqcup % d", x[j]);
    x[j] = d[j] + d[j];
  }
  reduce();
  printf("\nThe\_double-d\_vector\_reduces\_to");
  for (j = 1; j < n; j ++) {
    printf(" " " ", x[j]);
    x[j] = d[j] + d[j] - x[j];
  reduce();
  printf(\verb"\n_and_the_zero_vector_is");
  for (j = 1; j < n; j ++) {
    printf(" " " ", x[j]);
  printf("\n");
This code is used in section 1.
```

4 INDEX SAND §6

6. Index.

Arc: 1. arcs: 3. $argc: \underline{1}, \underline{2}.$ $argv: \ \underline{1}, \ \underline{2}.$ $d: \underline{1}.$ exit: 2.fprintf: 2.Graph: 1. $h: \underline{4}.$ $j: \underline{1}, \underline{4}.$ $k: \underline{1}, \underline{4}.$ $main: \underline{1}.$ $n: \underline{1}.$ next: 3. $panic_code$: 2. printf: 5.r: $\underline{1}$. reduce: $\underline{4}$, $\underline{5}$. $restore_graph$: 2. sscanf: 2. stderr: 2.t: $\underline{\mathbf{1}}$. $tip: \overline{}$ 3. $vec: \underline{1}, 3, 4.$ Vertex: 1.vertices: 3.

 $x: \underline{1}.$

SAND NAMES OF THE SECTIONS 5

```
 \begin{array}{ll} \left\langle \text{Input the graph 2} \right\rangle & \text{Used in section 1.} \\ \left\langle \text{Prepare the } \textit{vec} \text{ table 3} \right\rangle & \text{Used in section 1.} \\ \left\langle \text{Reduce the vector } d \ 5 \right\rangle & \text{Used in section 1.} \\ \left\langle \text{Subroutines 4} \right\rangle & \text{Used in section 1.} \end{array}
```

SAND

	Section	Page
Intro	 1	1
Indov	6	,