1* Intro.   Given a graph g with m edges, make data from which DLX2 should tell us all ways to label the vertices, using distinct labels in $\{0, 1, \ldots, m\}$, so that the edges have distinct difference. (Those differences will be $\{1, \ldots, m\}$.

Each label could be complemented with respect to m. I avoid this by "orienting" the edge labeled m.

#define $encode(x)$ $((x) < 10 ? (x) + \text{'0'} : (x) < 36 ? (x) - 10 + \text{'a'} : (x) < 62 ? (x) - 36 + \text{'A'} : (x) + 99)$
#define $maxm$ 156   /∗ based on that encoding, but I could go higher in a pinch! ∗/

#include `<stdio.h>`
#include `<stdlib.h>`
#include `"gb_graph.h"`
#include `"gb_save.h"`
  int c;

  $main(\textbf{int} \ argc, \textbf{char} \ *argv[\,])$
  {
    register int i, j, k, m, n;
    register Arc $*a$;
    register Graph $*g$;
    register Vertex $*v$;

    ⟨ Process the command line 2 ⟩;
    ⟨ Output the item-name line 3* ⟩;
    for $(k = 1; \ k \le m; \ k{+}{+})$ ⟨ Output the options for edge k 4 ⟩;
    for $(v = g\text{-}vertices; \ v < g\text{-}vertices + n; \ v{+}{+})$ ⟨ Output the options for vertex v 5* ⟩;
  }

2.   ⟨ Process the command line 2 ⟩ ≡
  if $(argc \ne 2)$ {
    $fprintf(stderr, \texttt{"Usage:\textvisiblespace\%s\textvisiblespace foo.gb\textbackslash n"}, argv[0])$;
    $exit(-1)$;
  }
  $g = restore\_graph(argv[1])$;
  if $(\neg g)$ {
    $fprintf(stderr, \texttt{"I\textvisiblespace couldn't\textvisiblespace reconstruct\textvisiblespace graph\textvisiblespace\%s!\textbackslash n"}, argv[1])$;
    $exit(-2)$;
  }
  $m = g\text{-}m/2, n = g\text{-}n$;
  if $(m \ge maxm)$ {
    $fprintf(stderr, \texttt{"Sorry,\textvisiblespace at\textvisiblespace present\textvisiblespace I\textvisiblespace require\textvisiblespace m<\%d!\textbackslash n"}, maxm)$;
    $exit(-3)$;
  }
  $printf(\texttt{"|\textvisiblespace\%s\textvisiblespace\%s\textbackslash n"}, argv[0], argv[1])$;
This code is used in section 1*.

3* There's a primary item k for each edge label, and a primary item uv for each edge. This enforces a permutation between edges and labels.

This version also introduces a primary item `#`v for each vertex.

There's a secondary item `.`v for each vertex; its color will be its label.

There's a secondary item `+`k for each vertex label; its color will be the vertex so labeled.

⟨ Output the item-name line 3* ⟩ ≡

  for $(k = 1;\ k \leq m;\ k{+}{+})$ $printf$ (`"%c␣"`, $encode(k)$);

  for $(v = g{\rightarrow}vertices;\ v < g{\rightarrow}vertices + n;\ v{+}{+})$

    for $(a = v{\rightarrow}arcs;\ a;\ a = a{\rightarrow}next)$

      if $(a{\rightarrow}tip > v)$ $printf$ (`"%s-%s␣"`, $v{\rightarrow}name$, $a{\rightarrow}tip{\rightarrow}name$);

  for $(v = g{\rightarrow}vertices;\ v < g{\rightarrow}vertices + n;\ v{+}{+})$ $printf$ (`"#%s␣"`, $v{\rightarrow}name$);

  $printf$ (`"|"`);

  for $(v = g{\rightarrow}vertices;\ v < g{\rightarrow}vertices + n;\ v{+}{+})$ $printf$ (`"␣.%s"`, $v{\rightarrow}name$);

  for $(k = 0;\ k \leq m;\ k{+}{+})$ $printf$ (`"␣+%c"`, $encode(k)$);

  $printf$ (`"\n"`);

This code is used in section 1*.

4. #**define** $vrt(v)$ $((\mathbf{int})((v) - g{\rightarrow}vertices))$

⟨ Output the options for edge k 4 ⟩ ≡

  {

    for $(i = 0, j = k;\ j \leq m;\ i{+}{+}, j{+}{+})$ {

      for $(v = g{\rightarrow}vertices;\ v < g{\rightarrow}vertices + n;\ v{+}{+})$

        for $(a = v{\rightarrow}arcs;\ a;\ a = a{\rightarrow}next)$

          if $(a{\rightarrow}tip > v)$ {

            $printf$ (`"%c␣%s-%s␣.%s:%c␣.%s:%c␣+%c:%c␣+%c:%c\n"`, $encode(k)$, $v{\rightarrow}name$, $a{\rightarrow}tip{\rightarrow}name$,

                $v{\rightarrow}name$, $encode(i)$, $a{\rightarrow}tip{\rightarrow}name$, $encode(j)$, $encode(i)$, $encode(vrt(v))$, $encode(j)$,

                $encode(vrt(a{\rightarrow}tip)))$;

            if $(i \neq 0 \vee j \neq m)$ /∗ prevent complementation symmetry ∗/

              $printf$ (`"%c␣%s-%s␣.%s:%c␣.%s:%c␣+%c:%c␣+%c:%c\n"`, $encode(k)$, $v{\rightarrow}name$, $a{\rightarrow}tip{\rightarrow}name$,

                  $v{\rightarrow}name$, $encode(j)$, $a{\rightarrow}tip{\rightarrow}name$, $encode(i)$, $encode(j)$, $encode(vrt(v))$, $encode(i)$,

                  $encode(vrt(a{\rightarrow}tip)))$;

          }

    }

  }

This code is used in section 1*.

5* ⟨ Output the options for vertex v 5* ⟩ ≡

  {

    for $(k = 0;\ k \leq m;\ k{+}{+})$ $printf$ (`"#%s␣.%s:%c␣+%c:%c\n"`, $v{\rightarrow}name$, $v{\rightarrow}name$, $encode(k)$, $encode(k)$,

      $encode((\mathbf{int})(v - g{\rightarrow}vertices)))$;

  }

This code is used in section 1*.

6*. Index.

The following sections were changed by the change file: 1, 3, 5, 6.

$\langle$ Output the item-name line 3^* $\rangle$ Used in section 1^*.
$\langle$ Output the options for edge k 4 $\rangle$ Used in section 1^*.
$\langle$ Output the options for vertex v 5^* $\rangle$ Used in section 1^*.
$\langle$ Process the command line 2 $\rangle$ Used in section 1^*.

GRACEFUL-DLX-DOMAINS