1. Intro. A simple program to make "random" squaregraphs, by sort of a "crocheting" technique. (Hacked in haste.) #define maxn 1000 #include <stdio.h> #include <stdlib.h> #include "gb_flip.h" int a[2 * maxn + 4], d[2 * maxn + 8];int move[8*maxn];int count[maxn]; int seed; int steps; main(int argc, char *argv[]) register int j, k, m, t, w; $\langle \text{Process the command line } 2 \rangle$; a[0] = 0, a[1] = 1, a[2] = 0, a[3] = 1;d[0] = d[1] = d[2] = d[3] = 2;w = 4;for (j = 0; j < steps; j++) { $\langle \text{ Set } m \text{ to the number of possible moves } 3 \rangle;$ $k = gb_unif_rand(m);$ $\langle \text{ Make move } k \mid 4 \rangle;$ $\langle \text{ Check for pairs 5} \rangle;$ $\langle \text{Output the result 6} \rangle$; $\langle \text{Process the command line } 2 \rangle \equiv$ if $(argc \neq 3 \lor sscanf(argv[1], "%d", \&steps) \neq 1 \lor sscanf(argv[2], "%d", \&steps) \neq 1)$ { fprintf(stderr, "Usage: "%s = n = seed n", argv[0]);exit(-1); if $(steps \ge maxn)$ { $fprintf(stderr, "Sorry, _n_should_be_less_than_%d! n", maxn);$ exit(-2); $gb_init_rand(seed);$ This code is used in section 1. 3. $\langle \text{Set } m \text{ to the number of possible moves } 3 \rangle \equiv$ d[w] = d[0], d[w+1] = d[1], a[w] = a[0], a[w+1] = a[1];for (m = 0; m < w; m++) move[m] = m;for (k = 0; k < w; k++)**if** (d[k+1] > 3) move[m++] = maxn + k;for (k = 0; k < w; k++)if $(d[k+1] > 3 \land d[k+2] > 3)$ move [m++] = maxn + maxn + k; This code is used in section 1.

```
4. \langle \text{ Make move } k | \mathbf{4} \rangle \equiv
  if (move[k] < maxn) {
     w += 2, k = move[k];
     for (m = w - 1; m \ge k + 2; m - 1) d[m + 2] = d[m], a[m + 2] = a[m];
     d[k+3] = d[k+1] + 1, d[k+2] = d[k+1] = 2, d[k] = d[k] + 1;
     a[k+3] = a[k+1], a[k+2] = j+2, a[k+1] = a[k], a[k] = j+2;
    if (k+3 \ge w)
       for (t = 0; t + w \le k + 3; t++) d[t] = d[w + t], a[t] = a[w + t];
  } else if (move[k] < maxn + maxn) {
     k = move[k] - maxn;
     d[k+1] = 2;
     t = a[k+1], a[k+1] = a[k], a[k] = t;
    if (k + 1 \ge w)
       for (t = 0; t + w \le k + 1; t + ) d[t] = d[w + t], a[t] = a[w + t];
  } else {
     k = move[k] - maxn - maxn;
     for (t = 0; t < w; t++)
       if (a[t] \equiv a[k+2] \land t \neq k+2) \ a[t] = a[k];
     a[k] = a[k+1], a[k+1] = a[k+3], d[k] = d[k] + 1, d[k+1] = d[k+3] + 1;
     for (t = k + 2; t < w; t++) a[t] = a[t + 2], d[t] = d[t + 2];
This code is used in section 1.
5. \langle \text{ Check for pairs 5} \rangle \equiv
  for (k = 0; k < j + 2; k++) count[k] = 0;
  for (k = 0; k < w; k++) count[a[k]]++;
  for (k = 0; k < j + 2; k++)
     if (count[k] \neq 0 \land count[k] \neq 2) fprintf (stderr, "count[%d] \sqcup is \sqcup %d! \n", k, count[k]);
This code is used in section 1.
6. \langle \text{ Output the result } 6 \rangle \equiv
  for (k = 0; k < w; k++) {
     printf(" " " d", a[k]);
     if (k \% 20 \equiv 19) printf("\n");
  if (k \% 20 \neq 0) printf("\n");
This code is used in section 1.
```

7. Index.

```
a: \underline{1}.
argc: \underline{1}, \underline{2}.
argv: \underline{1}, \underline{2}.
count: 1, 5.
d: \underline{\mathbf{1}}.
exit: 2.
fprintf: 2, 5.
gb\_init\_rand: 2.
gb\_unif\_rand: 1.
j: \underline{1}.
k: \underline{1}.
m: \underline{1}.
main: \underline{1}.
maxn\colon \ \underline{1},\ \underline{2},\ \underline{3},\ \underline{4}.
move: \underline{1}, 3, 4.
printf: 6.
seed: \underline{1}, \underline{2}.
sscanf: 2.
\begin{array}{ccc} stderr : & 2, & 5. \\ steps : & \underline{1}, & 2. \end{array}
t: \underline{1}.
```

 $w: \underline{1}.$

4 NAMES OF THE SECTIONS

 ${\bf SQUAREGRAPH\text{-}RAND}$

```
\begin{array}{ll} \langle \, {\rm Check \ for \ pairs \ 5} \, \rangle & {\rm Used \ in \ section \ 1.} \\ \langle \, {\rm Make \ move \ } k \ 4 \, \rangle & {\rm Used \ in \ section \ 1.} \\ \langle \, {\rm Output \ the \ result \ 6} \, \rangle & {\rm Used \ in \ section \ 1.} \\ \langle \, {\rm Process \ the \ command \ line \ } 2 \, \rangle & {\rm Used \ in \ section \ 1.} \\ \langle \, {\rm Set \ } m \ to \ the \ number \ of \ possible \ moves \ 3} \, \rangle & {\rm Used \ in \ section \ 1.} \end{array}
```

SQUAREGRAPH-RAND

	Section	Pag	56
Intro]
Indev	7		•