

## Tous les arbres couvrants d'un graphe complet

cf. livre *Combien ? tome 3 : graphes , chapitre 3, page 80.*

Il s'agit de l'application du codage de Prüfer permettant d'obtenir tous les arbres à nœuds numérotés. Voici le programme complet :

```
#include <SDL/SDL.h>
#include <math.h>
#define deuxpi 6.28318
#define rayon 20
#define N 5
void pause(void);
void putpixel(int xe, int ye, Uint32 couleur);
Uint32 getpixel(int xe, int ye);
void cercle(int x0, int y0, int R, Uint32 couleur);
void ligne(int x0, int y0, int x1, int y1, Uint32 c);
SDL_Surface *ecran = NULL;
Uint32 blanc, noir, rouge;
int xorig, yorig;
int x[N+1], y[N+1], file[N+1];

int main(int argc, char **argv)
{
int m[N-1], deg[N+1];
int nombre, i, j, k, r, q, etape;
xorig=rayon+10;
yorig=rayon+10;
SDL_Init(SDL_INIT_VIDEO);
ecran = SDL_SetVideoMode(800,600,32,SDL_HWSURFACE|SDL_DOUBLEBUF);
blanc = SDL_MapRGB(ecran->format,255,255,255);
noir = SDL_MapRGB(ecran->format,0,0,0);
rouge = SDL_MapRGB(ecran->format,255,0,0);
SDL_FillRect(ecran,0,blanc);

for(nombre=0;nombre<pow(N,N-2); nombre++)
{
for(i=1; i<=N;i++)
{ x[i]=xorig+rayon*cos(deuxpi*(i-1)/(float)N);
y[i]=yorig-rayon*sin(deuxpi*(i-1)/(float)N);
cercle(x[i],y[i],2,noir);
}
q = nombre;
for(i=0;i<N-2;i++) { r = q%N; m[N-2-i] = r+1; q = q/N; } /* m[] rempli de la case
1 à N-2 */

for(i=1;i<=N;i++) deg[i] = 1;
for(j=1;j<=N-2;j++) deg[m[j]]++;
```

```

for(etape=1;etape<=N-1;etape++)
{
    k=1;
    for(j=1;j<=N;j++) if(deg[j] == 1) {file[k] = j; k++;}
    if(etape < N-1)
    {
        ligne(x[file[1]],y[file[1]],x[m[etape]],y[m[etape]],rouge);
        deg[file[1]]--;
        deg[m[etape]]--;
    }
    else ligne(x[file[1]],y[file[1]],x[file[2]],y[file[2]],rouge);
}

xorig+=2*rayon+40; if(xorig>750){ yorig+=2*rayon+20; xorig=rayon+10; }
if(yorig>550) { SDL_Flip(ecran); pause(); SDL_FillRect(ecran,0,blanc);
                xorig=10+rayon; yorig=10+rayon;
            }
}
SDL_Flip(ecran);pause();return 0;
}

void pause(void)
{
    SDL_Event evenement;
    do SDL_WaitEvent(&evenement);
    while (evenement.type != SDL_QUIT && evenement.type != SDL_KEYDOWN);
}

void putpixel(int xe, int ye, Uint32 couleur)
{ Uint32 * numerocase;
  numerocase= (Uint32 *)(ecran->pixels)+xe+ye*ecran->w; *numerocase=couleur;
}

Uint32 getpixel(int xe, int ye)
{ Uint32 * numerocase;
  numerocase= (Uint32 *)(ecran->pixels)+xe+ye*ecran->w; return (*numerocase);
}

void ligne(int x0,int y0, int x1,int y1, Uint32 c)
{
    int dx,dy,x,y,residu,absdx,absdy,pasx,pasy,i;
    dx=x1-x0; dy=y1-y0; residu=0; x=x0;y=y0; putpixel(x,y,c);
    if (dx>0) pasx=1;else pasx=-1; if (dy>0) pasy=1; else pasy=-1;
    absdx=abs(dx);absdy=abs(dy);
    if (dx==0) for(i=0;i<absdy;i++) { y+=pasy;
                                    putpixel(x,y,c); /*cercle(x,y,3,c);*/}
    else if(dy==0) for(i=0;i<absdx;i++){ x+=pasx;
                                    putpixel(x,y,c); /*cercle(x,y,3,c);*/}
    else if (absdx==absdy)

```

```

for(i=0;i<absdx;i++) { x+=pasx; y+=pasy;
    putpixel(x,y,c);
}
else if (absdx>absdy)
    for(i=0;i<absdx;i++)
        { x+=pasx; residu+=absdy;
          if(residu >= absdx) {residu -=absdx; y+=pasy;}
          putpixel(x,y,c);/*cercle(x,y,3,c);*/
        }
else for(i=0;i<absdy;i++)
    {y+=pasy; residu +=absdx;
    if (residu>=absdy) {residu -= absdy;x +=pasx;}
    putpixel(x,y,c);/*cercle(x,y,3,c);*/
    }
}

void cercle( int xo, int yo, int RR, Uint32 couleur)
{
    int x, y, F, F1, F2,newx,newy;
    x=xo; y=yo+RR; F=0;
    if (x<800 && x>=0 && y>=0 && y<600) putpixel(x,y,couleur);
    if (x<800 && x>=0 && 2*yo-y>=0 && 2*yo-y<600) putpixel (x,2*yo-y, couleur);
    while ( y>yo)
    {
        F1=F+2*(x-xo)+1;F2=F-2*(y-yo)+1;
        if ( abs(F1)<abs(F2)) { x+=1; F=F1; }
        else{ y-=1; F=F2; }
        if (x<800 && x>=0 && y>=0 && y<600) putpixel(x,y,couleur);
        newx=2*xo-x ; newy=2*yo-y ;
        if (x<800 && x>=0 && newy>=0 && newy<600) putpixel(x, newy,couleur);
        if (newx<800 && newx>=0 && y>=0 && y<600) putpixel( newx,y,couleur);
        if (newx<800 && newx>=0 && newy>=0 && newy<600) putpixel(newx,
            newy, couleur);
    }
    if (xo+RR<800 && xo+RR>=0) putpixel(xo+RR,yo,couleur);
    if (xo-RR<800 && xo-RR>=0) putpixel(xo-RR,yo, couleur);
}

```