

Programa 5.32 Estruturas de dados

```

const
  MAXNUMVERTICES = 100000; {—No. maximo de vertices—}
  MAXNUMARESTAS = 100000; {—No. maximo de arestas—}
  MAXR = 5;
  MAXTAMPROX = MAXR*MAXNUMARESTAS;
  MAXTAM = 1000; {—Usado Fila—}
  MAXTAMCHAVE = 6; {—No. maximo de caracteres da chave—}
  MAXNUMCHAVES = 100000; {—No. maximo de chaves lidas—}
  INDEFINIDO = -1;

type
  {—Tipos usados em GrafoListaInc do Programa 7.25 —}
  TipoValorVertice = -1..MAXNUMVERTICES;
  TipoValorAresta = 0..MAXNUMARESTAS;
  Tipor = 0..MAXR;
  TipoMaxTamProx = -1..MAXTAMPROX;
  TipoPesoAresta = integer;
  TipoArranjoVertices = array[Tipor] of TipoValorVertice;
  TipoAresta = record
    Vertices: TipoArranjoVertices;
    Peso: TipoPesoAresta;
  end;
  TipoArranjoArestas = array[TipoValorAresta] of TipoAresta;
  TipoGrafo =
    record
      Arestas: TipoArranjoArestas;
      Prim: array[TipoValorVertice] of TipoMaxTamProx;
      Prox: array[TipoMaxTamProx] of TipoMaxTamProx;
      ProxDisponivel: TipoMaxTamProx;
      NumVertices: TipoValorVertice;
      NumArestas: TipoValorAresta;
      r: Tipor;
    end;
  TipoApontador = integer;
  {—Tipos usados em Fila do Programa 3.17 —}
  TipoItem = record
    Chave: TipoValorVertice;
    { outros componentes }
  end;
  TipoFila = record
    Item: array[1..MaxTam] of TipoItem;
    Frente: TipoApontador;
    Tras: TipoApontador;
  end;
  TipoPesos = array[1..MAXTAMCHAVE] of integer;
  TipoTodosPesos = array[Tipor] of TipoPesos;
  Tipog = array[0..MAXNUMVERTICES] of integer;
  TipoChave = packed array[1..MAXTAMCHAVE] of char;
  TipoConjChaves = array[0..MAXNUMCHAVES] of TipoChave;
  TipoIndice = TipoValorVertice;

```

O Programa 5.33 gera um grafo sem *selfloops* e sem arestas repetidas, a partir de r vértices obtidos mediante chamadas sucessivas a r funções de transformação h_0, h_1, \dots, h_{r-1} universais. O Programa 5.34 apresenta o refinamento final para obter uma função de transformação perfeita. A implementação da função de transformação perfeita para ser usada em dicionários está descrita no Programa 5.35 e o Programa 5.36 testa o funcionamento da função perfeita.

Programa 5.33 Gera um grafo sem arestas repetidas e sem selfloops

```

procedure GeraGrafo (var ConjChaves: TipoConjChaves;
  N: TipoValorAresta;
  M: TipoValorVertice;
  r: Tipor;
  var Pesos: TipoTodosPesos;
  var NgrafosGerados: integer;
  var Grafo: TipoGrafo);
{ Gera um grafo sem arestas repetidas e sem selfloops }
var i, j: integer; Aresta: TipoAresta; GrafoValido: boolean;

function VerticesIguais (Aresta: TipoAresta): boolean;
var i, j: integer;
begin
  VerticesIguais := false;
  for i := 0 to Grafo.r - 2 do
    for j := i + 1 to Grafo.r - 1 do
      if Aresta.Vertices[i] = Aresta.Vertices[j]
      then VerticesIguais := true;
    end;
  end;

begin { —GeraGrafo— }
  repeat
    GrafoValido := true; Grafo.NumVertices := M;
    Grafo.NumArestas := N; Grafo.r := r;
    FGVazio (Grafo); NgrafosGerados := 0;
    for j := 0 to Grafo.r - 1 do GeraPesos (Pesos[j]);
    for i := 0 to Grafo.NumArestas - 1 do
      begin
        Aresta.Peso := i;
        for j := 0 to Grafo.r - 1 do
          Aresta.Vertices[j] := h (ConjChaves[i], Pesos[j]);
        end;
        if VerticesIguais (Aresta) or ExisteAresta (Aresta, Grafo)
        then begin GrafoValido := false; break; end
        else InsereAresta (Aresta, Grafo);
      end;
    NgrafosGerados := NgrafosGerados + 1;
  until GrafoValido;
end; { GeraGrafo }

```