41 画像の拡大縮小

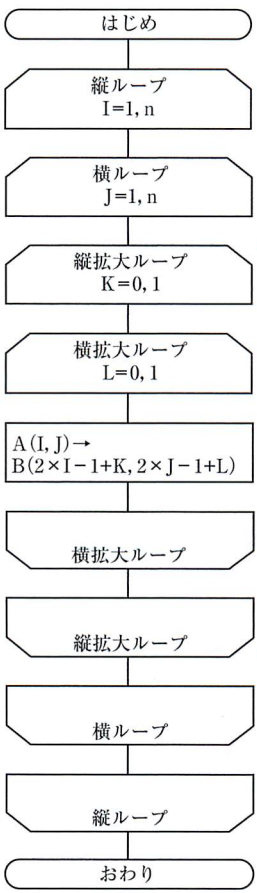
1 アルゴリズムの概要

- n×nで表されたビットマップ画像を，拡大縮小する。

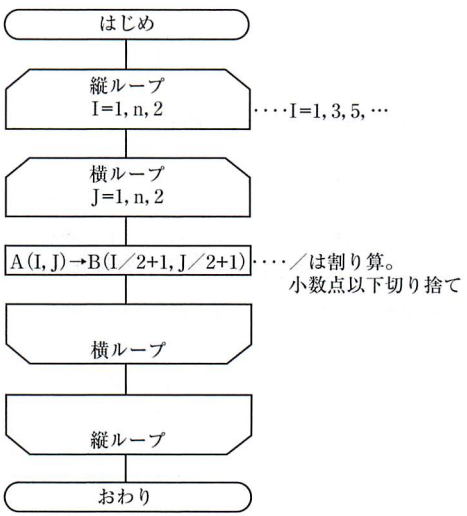
2 流れ図

- 元画像の縦i番，横j番の点をA（i，j）で，出力画像の縦i番，横j番の点をB（i，j）で表す。

〈①画像の拡大〉



〈②画像の縮小〉

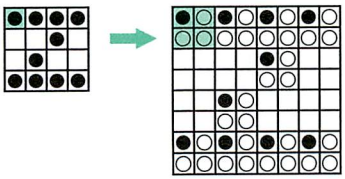


（注）説明文ではi，jでも，流れ図ではI，Jが使われることが試験問題でもよくあります。変数名に英大文字しか使えないプログラム言語があったからです。

3 拡大縮小の様子

①2倍拡大

Aの1点を4点にしてBに出力する。

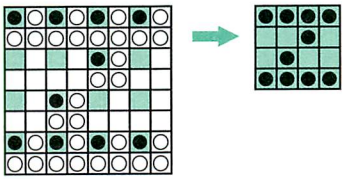


- A (i, j) を次の4点に転送する。
  B (2i－1, 2j－1)
  B (2i－1, 2j)
  B (2i, 2j－1)
  B (2i, 2j)

(例) A (1, 1) は，次の4点に転送される。
B (1, 1), B (1, 2), B (2, 1), B (2, 2)

②1／2縮小

縦横が奇数の点だけをBに出力する。



- A (i, j) のi, j＝1, 3, 5, 7…の点だけを，B (i/2＋1, j/2＋1) に転送する。

(例) A (7, 7) は，
$i_B = 7/2+1 = 4$
$j_B = 7/2+1 = 4$
で，B (4, 4) に転送される。

1 画像は，点の集まりで構成されている

画像は，**ピクセル**という点の集まりで構成されています。この点をA (i, j) で表すと，白黒画像の場合には0か1が，カラー画像の場合には色を表す数値が格納されています。点A (i, j) をそのまま転送すれば画像をコピーできます。iに5を加えて転送すれば下方向へ，jに5加えて転送すれば右方向へ5移動します。

Point この種のアルゴリズムは，具体的な例を考え，画像の四隅と中央付近の点がどこへ移動するかを考えてみると，穴埋め問題も簡単に解けます。

2 拡大するとギザギザになる

拡大の流れ図のように，A (i, j) を4点に転送すると拡大することができますが，点ごとに拡大する，つまり，点が大きくなるため，拡大された画像はギザギザ（ジャギー）になります。一昔前のワープロは，文字を16×16程度の点で表していて，拡大するとギザギザになっていました。

拡大縮小の流れ図を修正すると3倍拡大や3分の1縮小なども行うことができます。考えてみてください。

Memo 画像変換処理に関する出題が増えていますが，ここで示した流れ図同様，ビットマップを2次元配列に置き換えたものが多く，実際には2次元配列の問題になっています。