

メディアプログラミング演習—第8回（第4テーマ1日目）—

第4テーマ：領域埋めアルゴリズム（ペイント処理）

平面上の複雑な形をした領域を「塗りつぶす」アルゴリズムを扱う。本実習で言う「領域」とは、「画像の中心を含む、黒色のピクセルがつながってできている閉領域」と想定する。従って、この閉領域内の全てのピクセルを「黒」にするアルゴリズムを稼働化するのがテーマその4である。例を示せば、以下の通りである。

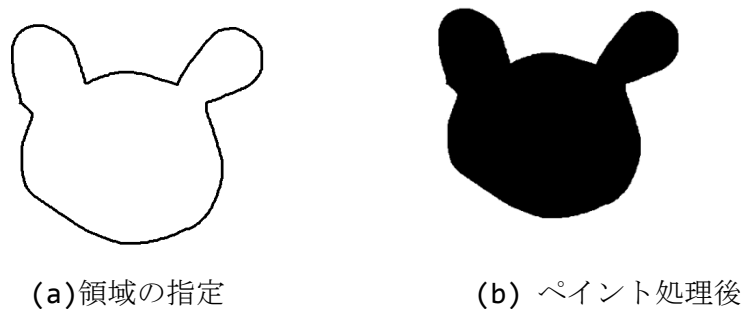


図8-1 ペイント処理

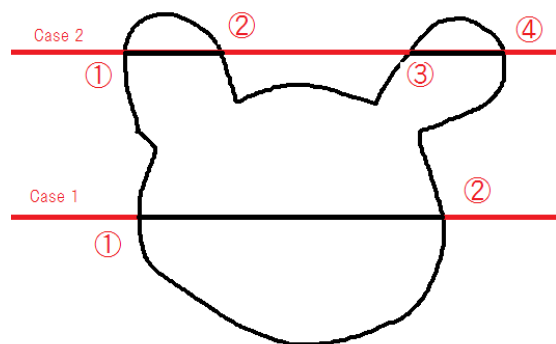
これを行う方法としては幾つかあるが、ここでは次の2つを扱う。

- (a) 走査線ごとにペイント対象の領域の外であるか中であるかを判断し、内部のピクセルを塗る方法。
- (b) 領域内の任意の点の、近傍（4または8ピクセル）を調べ、ペイントされていないピクセルがあればそれを塗り、すべての点を塗りつぶすまで繰り返す方法。

演習8（1）：スキャンラインによる塗りつぶし

前述の(a)の方法である。図8-2を用いて説明する。これは、横方向（縦方向でも同じである）のピクセルの列（スキャンラインと呼ぶ）を対象として、領域との境目を見出し、その間を塗り潰す方法である。図中 **Case1** では、左から順にピクセルを走査し、境界との交点①を見出す。この点から、同方向に、境界との交点②まで塗る。また、**Case2** にあるように境界と複数交わる場合は、『奇数番目の交点から偶数番目の交点の間を塗る』こととする。これらを、最上位のラインから最下位のラインまで繰り返すことにより、領域を塗り潰すことができる。

図8-2 スキャンライン法



アルゴリズム：アルゴリズムの概要は以下の通りとなる。

```

y=0;
while( y<img_in.height ){
  x=0;
  while(x<img_in.width){
    while( ( x<img_in.width)&&(Is_White(x,y) ==1 )) { x++; }; ①
    while( ( x<img_in.width)&&(Is_White(x,y) ==0 )) { x++; }; ②
    while( ( x<img_in.width)&&(Is_White(x,y) ==1 ))
      { Set_Black(x,y); x++; }; ③
    while( ( x<img_in.width)&&(Is_White(x,y) ==0 )) { x++; }; ④
  };
  y++;};

```

ピクセル(x,y)が白ならば「1」を返す

ピクセル(x,y)を黒に

処理内容は以下の通りである。

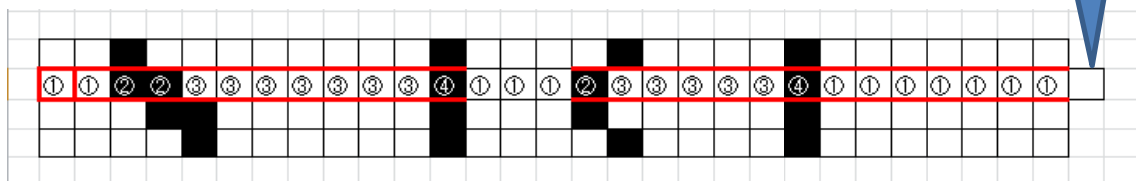
```

以下を「全てのライン」で繰り返す
{ 以下を「ラインの終わりまで」繰り返す
  { *①「白」をスキップ
    *②次いで「黒」をスキップ
    *③「白」である間、右方向に白を黒に換える
    *④「黒」をスキップ
  }
}

```

x=img.width となり、①の終了後、②③④をスキップ

ピクセルと上記①～④のどの段階で走査されるかを以下に示す。



演習 8-1. (a)プログラム sample4-1 の「 // この部分を作成 」の部分を作成させ、sample4-1 の画像で実行しなさい。

(b) 「ペイント」等で、任意の図形を描き、実行させなさい。

演習 8-2. 上記で、ライン上の白黒パターンがどのような場合に正しく動作しないかを説明しなさい。