令和４年６月７日

令和５年度大分大学理工学部３年次編入学試験問題【筆記試験】
（共創理工学科　知能情報システムコース）

試験問題の配点は，1.が 50 点，2.が 50 点です。

1. 次の英文を読んで各問いに答えなさい。

intervention: 介入	implementation: 導入	widespread: 普及した	entering: 入る
era: 時代	deployed: 展開する	promote: 促進する	bandwidth: 帯域幅
capture: 記録する	yield: もたらす		

(1) 下線部(A)の文章を和訳しなさい。
(2) 下線部(B)が指すものをすべて日本語で答えなさい。
(3) 下線部(C)となる要因を日本語で簡潔に答えなさい。
(4) 著者がエッジコンピューティングに期待することとその理由を日本語で答えなさい。

2．2次元平面上の点集合の凸包を求めるプログラムについて考える。凸包とは，図1に示すように，与えられた点集合内のすべての点を包含する最小の凸多角形のことである。このプログラムは，凸包の左端の頂点から，右回りで凸包の頂点を順番に取得していくものとする。与えられる点集合内に，xy 座標が全く同じ点が存在しないとした場合，以下の(1)～(5)の問いに答えなさい。

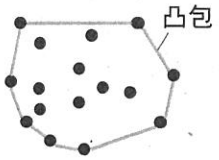

図1

(1) 図2において，点 p_i は凸包の頂点であることが確定しているものとする。このプログラムは，頂点である点 p_i に隣接する頂点の候補として点 p_{i1} が選ばれている場合，矢印 p_ip_{i1} の向きに対して左側に他の点が存在しないかという検証を繰り返す。左側に点が存在する場合は，その点を新しい頂点の候補点 p_{i1} とし，同様の検証を繰り返す。左側に他の点が存在しない場合は，頂点の候補点 p_{i1} を，頂点 p_i に隣接する凸包の頂点であると確定する。

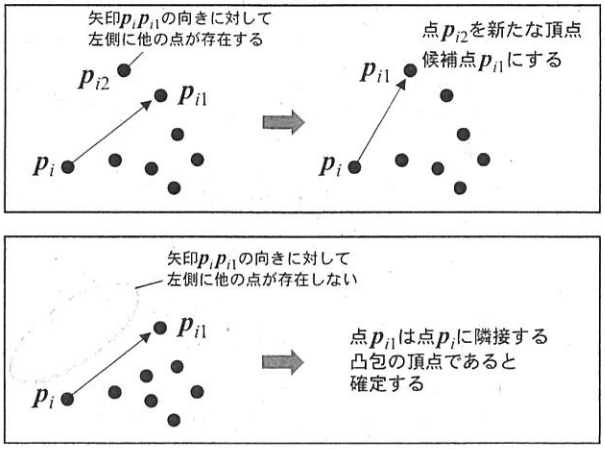

図2

図3に示すように，点 p_i と点 p_{i1} を結ぶベクトルを $\boldsymbol{a}=(a_x, a_y)$，点 p_i と点 p_{i2} を結ぶベクトルを $\boldsymbol{b}=(b_x, b_y)$ とする。この時，関数 $g(a_x, a_y, b_x, b_y) = a_x b_y - b_x a_y$ の値が正ならば，点 p_{i2} は矢印 p_ip_{i1} の向きに対して左側に存在し，負ならば点 p_{i2} は矢印 p_ip_{i1} の向きに対して右側に存在し，0ならば点 p_i，点 p_{i1}，点 p_{i2} は直線上に存在すると判定することができる。図4は，この関数 $g(a_x, a_y, b_x, b_y)$ をC言語で書いたものである。図4内の空白(a)を埋めなさい。

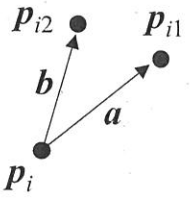

図3

```
/* ax, ay はベクトルaのx,y成分 */
/* bx, by はベクトルbのx,y成分 */
int g(int ax, int ay, int bx, int by) {
  return [    (a)    ];
}
```
図4

(2) このプログラムでは，最初に，必ず凸包の頂点となる点 p_0 を取得しておく必要がある。このような点 p_0 は，与えられた点集合の中から，x 座標が最小の点を求めることにより取得することができる。x 座標が最小の点が複数存在する場合は，それらの点の中から，さらに y 座標が最小の点を求めることにより，一意に取得することができる。図5は，このような点 p_0 の取得する関数をC言語で書いたものである。図5内の空白(b)～(d)を埋めなさい。

```
/* x[] は点集合のx座標列 */
/* y[] は点集合のy座標列 */
/* n   は点集合中の点の数 */
/* 戻り値は点p0に対応する配列x[],y[]内の要素番号 */
int start(int x[], int y[], int n) {
  int ret = 0, xmin, ymin, i;
  xmin = x[ret]; ymin = y[ret];
  for (i = 1; i < n; i++) {
    if ( [    (b)    ] ) {
      xmin = x[i]; ymin = y[i]; ret = i;
    }
    else if ( [    (c)    ] )
      if ( [    (d)    ] ) {
        ymin = y[i]; ret = i;
      }
  }
  return ret;
}
```
図5

(3) 図６は，図４，図５に示したＣ言語の関数を用いて，図７に示す点集合の凸包の頂点座標を出力するＣ言語で書かれたプログラムである。図６内の空白(e)～(g)を埋めなさい。

(4) 図６のプログラムを実行した場合，標準出力にどのような結果が表示されるか書きなさい。

(5) 図６のプログラムについて，図８に示す点集合の凸包の頂点座標を求めるために，x[3]の値を３から２に書き換えた場合，凸包の頂点(2,5)，(2,9)を結ぶ辺上に存在する点(2,6)が，頂点として取得されてしまう。この問題を解決するためには，図６中の条件式(h)について，変数 rf の値が正になった場合だけではなく，変数 rf の値が０になり，かつ，点 $p_{i}p_{i1}$ 間の距離よりも点 $p_{i}p_{i2}$ 間の距離の方が長い場合も，点 p_{i2} を新しい頂点の候補点 p_{i1} となるように判断させればよい。図９は，２点間の距離を求める関数をＣ言語で書いたものである。この関数を用いて，上記の問題が解決するように条件式(h)を修正しなさい。

```
#include <stdio.h>
#define N 10
int g(int ax, int ay, int bx, int by);
int start(int x[], int y[], int n);
double d(int x0, int y0, int x1, int y1);
int main() {
  int x[N] = { 9,7,5,3,2,2,8,8,5,7 };
  int y[N] = { 3,8,2,6,9,5,6,9,5,3 };
  int c[N], h = 0, pi, pi1, pi2, rf, i;
  pi = start( (e) );
  do {
    c[h] = pi; h++;
    pi1 = 0;
    for (pi2 = 1; pi2 < N; pi2++) {
      if (pi1 == pi)
        pi1 = pi2;
      else if (pi2 != pi) {
        rf = g( (f) );
        if (rf>0)
          pi1 = pi2;
      }
    }
    (g);
  } while (pi != c[0]);

  for (i = 0; i < h; i++)
    printf("(%d,%d)->", x[c[i]], y[c[i]]);
  return 0;
}
```

(h)

図６

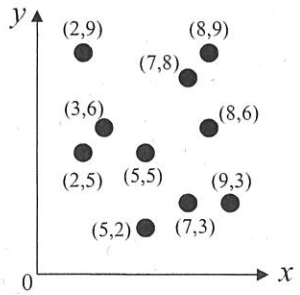

図７

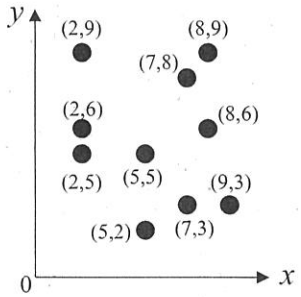

図８

```
#include <math.h>
/* x0, y0は点その１のxy座標 */
/* x1, y1は点その２のxy座標 */
double d(int x0, int y0, int x1, int y1) {
  return sqrt((x1 - x0)*(x1 - x0) + (y1 - y0)*(y1 - y0));
}
```

図９