筑波大学 情報学群 情報科学類・情報メディア創成学類

平成31年度 学群編入学試験

学力試験問題(専門科目)

[注意事項]

1. 試験開始の合図があるまで、問題の中を見てはいけません。
2. 解答用紙と下書き用紙の定められた欄に、氏名、受験番号を記入すること。
3. この問題冊子は全部で10ページ(表紙、白紙を除く)です。
4. 専門科目の選択について、

   (ア) 情報科学類と情報メディア創成学類を併願する者 は、問題 1 から問題 6(数学、情報基礎、物理学)の計 6 問から 4 問を選択して答えなさい。ただし、情報メディア創成学類の合否判定においては、数学と情報基礎の解答のみを評価します。

   (イ) 情報科学類を単願する者 は、問題 1 から問題 6(数学、情報基礎、物理学)の計 6 問から 4 問を選択して答えなさい。

   (ウ) 情報メディア創成学類を単願する者 は、問題 1 から問題 4(数学、情報基礎)の計 4 問をすべて答えなさい。

5. 解答用紙は、専門科目で選択した 4 問に対して、各問 1 枚の 合計 4 枚 を用いること。
6. 解答用紙上部の $\boxed{}$ 欄に解答する問題番号を記入すること。

問題1　数学(1)

(1) $x^2+y^2=4$ の条件の下で，$f(x,y)=4x+2xy$ の最小値，最大値を求めなさい．また，最小，最大となるときの x と y の値も示しなさい．

(2) 次の二重積分について，以下の問いに答えなさい．

$$V=\iint_D \left(x^2+xy\right)\ dxdy \qquad \cdots (*)$$

ただし，$D=\left\{(x,y)\mid x^2+y^2\leqq 1\right\}$ とする．

(2-1) x と y を極座標変換し，式 $(*)$ の右辺を書き換えなさい．

(2-2) V の値を求めなさい．

問題2　数学 (2)

(1) 零ベクトルでない m 次元の実列ベクトル $\boldsymbol{a}, \boldsymbol{b}$ について，$\boldsymbol{b}$ から $\boldsymbol{a}$ への射影を考える．$\boldsymbol{a}, \boldsymbol{b}$ の内積 $\boldsymbol{a}^T\boldsymbol{b}$ (ただし添字 T は転置を表す) および，$\boldsymbol{a}$ を正規化したベクトル $\frac{1}{\|\boldsymbol{a}\|}\boldsymbol{a}$ (ただし $\|\boldsymbol{a}\|$ はベクトル $\boldsymbol{a}$ のノルムを表す) を用いると，$\boldsymbol{b}$ から $\boldsymbol{a}$ への射影 $\boldsymbol{p}$ は

$$\boldsymbol{p} = \frac{\boldsymbol{a}^T\boldsymbol{b}}{\|\boldsymbol{a}\|^2}\boldsymbol{a}$$

と書ける．

(1-1) $\boldsymbol{p} = P\boldsymbol{b}$ となるような射影行列 P を求めなさい．

(1-2) $P^2 = P$ となることを示しなさい．

(1-3) I を単位行列としたとき，$I - P$ もまた射影行列となる．このとき，$\boldsymbol{b}$ を $I - P$ で射影して得られるベクトル $\boldsymbol{c}$ と，$\boldsymbol{a}, \boldsymbol{b}$ の関係を図示しなさい．

(2) 線形独立な n 個の m 次元実列ベクトル $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n$ (ただし $n < m$) によって張られる m 次元実数空間の部分空間を考え，この部分空間で，零ベクトルでない m 次元の実列ベクトル $\boldsymbol{b}$ に最も近いベクトル，すなわち射影 $\boldsymbol{q}$ を求めたい．実係数 $x_1, x_2, \ldots, x_n$ を用いて

$$\boldsymbol{q} = x_1\,\boldsymbol{a}_1 + x_2\,\boldsymbol{a}_2 + \cdots + x_n\boldsymbol{a}_n$$

とおく．さらに，

$$A = (\boldsymbol{a}_1\ \boldsymbol{a}_2\ \cdots\ \boldsymbol{a}_n),$$
$$\boldsymbol{x} = (x_1\ x_2\ \cdots\ x_n)^T$$

となるような $m\times n$ 行列 A および n 次元の実列ベクトル $\boldsymbol{x}$ を用いると，$\boldsymbol{q} = A\boldsymbol{x}$ と書ける．

(2-1) $\boldsymbol{q}$ から $\boldsymbol{b}$ に向かうベクトルと，$\boldsymbol{a}_i$ (ただし $i = 1, 2, \ldots, n$) によって張られる部分空間は直交する．この関係を表す式を，行列 A およびベクトル $\boldsymbol{b}, \boldsymbol{x}$ のみを用いて表しなさい．

(2-2) ベクトル $\boldsymbol{x}$ を，行列 A およびベクトル $\boldsymbol{b}$ のみを用いて表しなさい．

問題3　情報基礎（1）

横7マス，縦5マスの長方形の迷路において，「入口から各マスまでの距離」を計算するC言語のプログラムを考える．ここで，「入口からあるマスまでの距離」は，入口から上下左右の4方向のいずれかに1マスずつ進み，目的地のマスに到達する経路のうち，最も短い経路において進んだマスの数とする．ただし，経路は後述する「壁」を通らないものとする．

図1は迷路の例である．迷路の座標は，左上を $(0,0)$ として x 座標を右方向に，y 座標を下方向に取り，各マスの座標を非負の整数の組で表す．黒い四角は通れない壁を，白い四角は通れる通路を表す．たとえば入口が $(1,0)$ であるとき，入口から $(1,0)$ のマスまでの距離は0であり，入口から $(3,1)$ のマスまでの距離は7である．

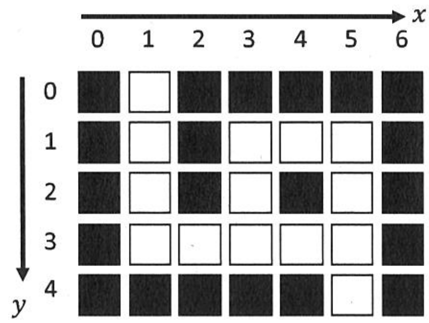

図1　迷路の座標系

図1の迷路に対応するプログラムの冒頭部分は以下の通りである．

```
#include <stdio.h>
#define X_SIZE 7
#define Y_SIZE 5
int maze[X_SIZE][Y_SIZE] = {
   {1, 1, 1, 1, 1},
   {0, 0, 0, 0, 1}, {1, 1, 1, 0, 1},
   {1, 0, 0, 0, 1}, {1, 0, 1, 0, 1},
   {1, 0, 0, 0, 0}, {1, 1, 1, 1, 1}
};
```

配列 maze は図1の迷路の配置を表す．maze[x][y] が1であれば，(x, y) のマスが壁であり，maze[x][y] が0であれば，(x, y) のマスが通路である．

次ページに続く

(1) 関数 ok は，整数 x と y が与えられたとき，配列 maze の表す迷路において，(x, y) が迷路内の通路の座標であるならば真 (0 以外の整数) を返し，それ以外であれば 0 を返す．以下の（ア）を埋めて，関数 ok を完成させなさい．

```
int ok(int x, int y) {
  [      (ア)      ]
}
```

(2) 入口から (x, y) までの距離を計算するため，次の配列 dist を用いる．

```
int dist[X_SIZE][Y_SIZE];
```

プログラムの実行の最初の段階では，dist[x][y] には十分大きな値を入れる．以下の（イ）を埋めて，配列 dist のすべての要素を X_SIZE * Y_SIZE という値に設定する関数 init_dist を完成させなさい．

```
void init_dist(void) {
  int x, y;
  [      (イ)      ]
}
```

(3) ここで採用するアルゴリズムは，入口からの距離が判明したマスから，隣接するマスに距離の情報を伝搬する操作を繰り返すことにより，入口からすべてのマスまでの距離を計算する．
次の関数 update_dist は，そのアルゴリズムの一部であり，(x1, y1) と (x2, y2) という隣接する 2 つのマスの座標が与えられた時，dist[x1][y1] の値を使って dist[x2][y2] の値を更新する．update_dist は，まず，(x1, y1) と (x2, y2) がともに迷路内の通路で，かつ，dist[x2][y2] の現在の値が dist[x1][y1] + 1 より大きければ dist[x2][y2] を dist[x1][y1] + 1 に書き換える．そうでない時は dist[x2][y2] を変更しない．関数 update_dist は，値の書き換えが起きた時 1 を返し，書き換えが起きなかった時 0 を返す．

```
int update_dist(int x1, int y1, int x2, int y2) {
  if ([      (ウ)      ]) {
    [      (エ)      ] ;
    return 1;
  }
  return 0;
}
```

次ページに続く

前述の（ウ），（エ）を埋めて関数 update_dist を完成させなさい．ただし，$(x1, y1)$ と $(x2, y2)$ が隣接することを前提としなさい．また，前問までに与えた関数を利用してよい．

(4) 次の関数 main は，これまで定義した配列や関数等を使って，入口から迷路のすべてのマスまでの距離を求めるものである．

```
int main(void) {
  int changed, x, y;
  init_dist();
  dist[1][0] = 0; /* スタート地点の距離を 0 にする */
  do {
    changed = 0;
    for (y = 0; y < Y_SIZE; y++) {
      for (x = 0; x < X_SIZE; x++) {
          ┌────────────┐
          │   （オ）   │
          └────────────┘
      }
    }
  } while (changed);
  for (y = 0; y < Y_SIZE; y++) {
    for (x = 0; x < X_SIZE; x++) {
      printf("%d,", dist[x][y]);
    }
    printf("\n");
  }
  return 0;
}
```

上記の（オ）には，(x, y) の上下左右に隣接する座標と (x, y) を引数として関数 update_dist を呼び出し，配列 dist の値に変化があった座標の数を changed に加える処理が入る．（オ）を埋めてプログラムを完成させなさい．

(5) 完成したプログラムを実行した際の出力を書きなさい．

(6) 関数 main に含まれる do...while のループは何回実行されるか答えなさい．

問題 4　情報基礎 (2)

文書や画像などの持つ特徴をベクトルの形で表現したものを特徴ベクトルと呼ぶ。ここでは、2 つの特徴ベクトル $\vec{x} = (x_1, \cdots, x_n)$ と $\vec{y} = (y_1, \cdots, y_n)$ の類似度として以下の定義（コサイン類似度）を用いる。ただし、n は 1 以上の整数とする。

$$S_C(\vec{x}, \vec{y}) = \frac{x_1y_1 + \cdots + x_ny_n}{\sqrt{{x_1}^2 + \cdots + {x_n}^2}\sqrt{{y_1}^2 + \cdots + {y_n}^2}}$$

S_C の値域は閉区間 $[-1, 1]$ である。得られた値が大きいほど、2 つのベクトルの間の類似度が高いと見なされる。

与えられた特徴ベクトルの集合の中から類似度の最も高いペアを求める C 言語のプログラム (2 ページ後に掲載) について考える。ただし、与えられる集合には、零ベクトルは含まれておらず、$S_C(\vec{x}, \vec{y}) = 1$ となるような $\vec{x}$ と $\vec{y}$ も含まれていないものとする。また、同一のベクトル間の類似度 (つまり、$S_C(\vec{x}, \vec{x})$) については考えないものとする。

(1) 3 つのベクトル $\vec{a} = (0.0, 1.0, 2.0)$, $\vec{b} = (1.0, 0.0, 2.0)$, $\vec{c} = (2.0, 1.0, 0.0)$ が与えられたとする。このとき、$\vec{b}$ と $\vec{c}$ のうち、どちらの方が $\vec{a}$ との類似度が高いかを答えなさい。

(2) プログラム中の関数 `in_prod` は、与えられた 2 つのベクトル $\vec{x}, \vec{y}$ の内積 $\vec{x} \cdot \vec{y} = x_1y_1 + \cdots + x_ny_n$ を求めるプログラムである。空欄（ア）を埋めて `in_prod` を完成させなさい。ただし、プログラムでは $\vec{x}$ を `double *v_a`、$\vec{y}$ を `double *v_b` でそれぞれ表す。

(3) プログラム中の関数 `cos_sim` は、前述の `in_prod` を利用して類似度を計算する関数である。空欄（イ）を埋めて `cos_sim` を完成させなさい。なお、平方根を求める際には関数 `sqrt` (double 型の値を引数とし、その非負の平方根を double 型で返す関数) を用いること。

(4) 入力として与えられるベクトルの数を m と置いたとき、すべてのペアの類似度を (3) の `cos_sim` を使って計算するために必要な `cos_sim` の最小の呼び出し回数を答えなさい。その際、交換法則 ($S_C(\vec{x}, \vec{y}) = S_C(\vec{y}, \vec{x})$) が成り立つことに留意すること。

次ページに続く

(5) プログラム中の関数 `main` は、特徴ベクトルの集合から得られるペアについて、`cos_sim` を使ってすべてのペアの類似度を計算し、最も類似度の高いペアを求める。空欄（ウ）を埋めてプログラムを完成させなさい。その際、以下の点に留意すること。

- `cos_sim` の呼び出し回数が最小となるようにすること。
- 入力のベクトル集合として配列 `inputs` を用いること。
- 結果の出力には空欄（ウ）の下にある `printf` の呼び出しが使われるようにすること。なお、仮に全く同じ類似度のペアが見つかった場合でも、表示されるのはそれらのペアのうちいずれか 1 つのみとする。

次ページに続く

プログラム

```
#include <stdio.h>
#include <math.h>

#define L_VEC 3 /* ベクトルの長さ */
#define N_VEC 100 /* ベクトルの数 */

/* 入力 */
double inputs[N_VEC][L_VEC] = {{0.0, 1.0, 2.0},
                               {1.0, 0.0, 2.0},
                               {2.0, 1.0, 0.0},
                               {1.0, 3.0, 1.0},
                               /* 途中省略 */
                               {1.0, 2.0, 3.0}};

/* 内積を計算する関数 */
double in_prod(double *v_a, double *v_b) {
  int i;
  double sum = 0.0;

                    (ア)

  return sum;
}

/* 類似度を計算する関数 */
double cos_sim(double *v_a, double *v_b) {
  return              (イ)              ;

}

/* メイン関数 */
int main(void) {
  int i, j;
  double sim;
  int best_pair[2];
  double best_value = -2.0; /* 最小値よりも小さい値で初期化 */

  /* 最も類似度の高いペアを探索 */

                    (ウ)

  /* 結果を出力 */
  printf("best: cos_sim(inputs[%d], inputs[%d]) = %f\n",
         best_pair[0], best_pair[1], best_value);

  return 0;
}
```

問題5　物理学（1）

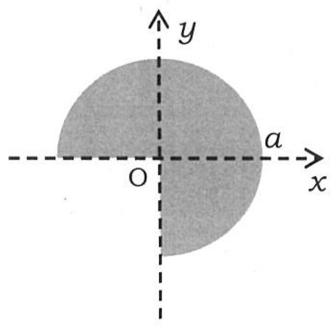

図1

図1のように、半径 a の円板から一部 $\left(\frac{1}{4}\right)$ を切り取った薄い板状の物体がある。ただし、板は厚みが無視でき、密度は一様で単位面積あたりの質量が σ である。x 軸と y 軸の交点を原点 O、重力加速度を g として、以下の問いに答えなさい。

(1) 質量中心の位置（x_G, y_G）を求めなさい。

(2) 原点 O を通り、xy 平面に垂直な z 軸まわりの慣性モーメント I_Z を求めなさい。

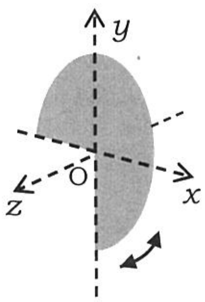

図2

次に、この物体を z 軸を固定軸として、そのまわりに滑らかに回転する振り子（図2のイメージ）を作り、物体を図3の状態から静かに離して回転させた。なお、重力は y 軸負方向に働いており、空気抵抗はないものとする。

(3) 物体が図4の状態となった瞬間の物体の角速度 ω の大きさを求めなさい。必要ならば、x_G, y_G, I_Z を用いて表しても良い。

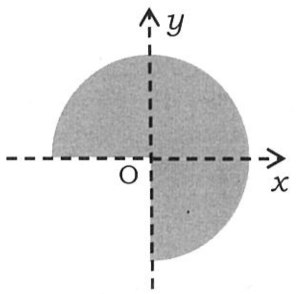

図3

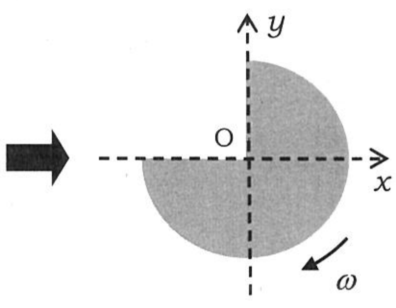

図4

問題6　物理学（2）

1. 図1のように、真空の空間にxyz座標をとる。点Aを $(x_0, 0, 0)$ の位置とし、そこに点電荷 q がある。yz平面上に無限に広くて薄い導体平板があり、接地されている。真空の誘電率を ε_0 として、次の問いに答えなさい。ただし、$x_0 > 0$ とする。
   (1) ある点 B に点電荷 $-q$ を置き、導体平板を取り去ったところ、$x > 0$の空間の電界は変わらなかった。点 B の座標を答えなさい。
   (2) (1)のあと、$(x, 0, 0)$ にある点 C での電位 ϕ を求めなさい。ただし、$x > x_0$とする。
   (3) $\frac{x_0}{x} = h$ を変数として、(2)で求めた電位 ϕ を、$h = 0$ のまわりでテイラー展開しなさい。ただし、x が x_0 より十分大きいとして、h の3次以上の項は無視するものとする。

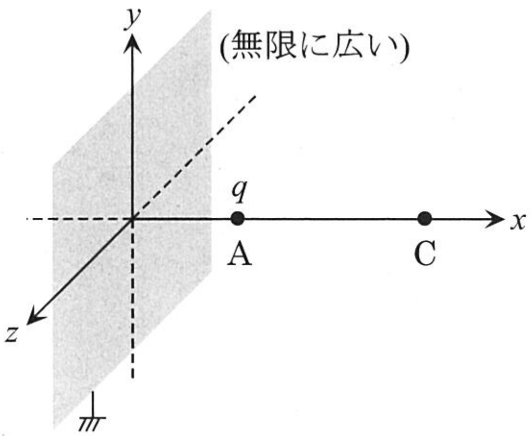

図1

2. 図2のように、x軸に中心軸を持ち、中心軸方向の長さがℓで、xにおける断面積 $S = S_0 + kx$、抵抗率 ρ の円錐台がある。下面Aで$x = 0$とする。下面Aと上面Bは、薄く電気抵抗がない電極である。A–B間の電気抵抗 R を求めなさい。ただし、$k \neq 0$, $S > 0$, $S_0 > 0$ とする。

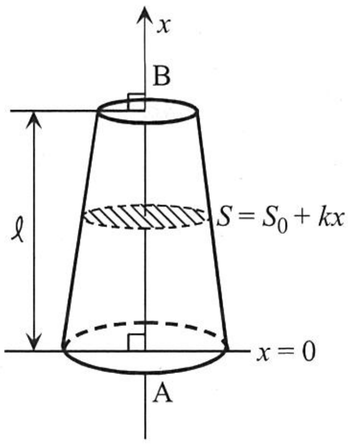

図2