問

２０２６年度　第２回
大学院情報科学研究科

修士課程
入学試験問題（一般）

[専門科目]

2026 年 2 月 20 日（金）
9:30～11:00

解　答　要　領

１．問題用紙と解答用紙の両方とも受験番号と氏名を必ず記入すること。

２．質問がある場合は，挙手をすること。

３．問題用紙，解答用紙とも提出すること。

４．問１～問３をすべて解答すること（解答は英語あるいは日本語いずれでも良い）。
　　解答は，それぞれ問題番号が書かれた解答用紙に解答すること。

５．解答用紙が足りない場合は試験監督に申し出ること。

６．すべて「参照・使用」不可。

受験番号	
氏　　名	

2026 年度第 2 回法政大学大学院情報科学研究科入学者選抜試験
問題用紙

試験科目	修士課程 （一般）
専門科目	

【問１】 $f(\boldsymbol{x}) = x_2^2 + 2x_1x_3 = 1$ で表される曲面がある．以下の問いに答えよ．

(1) $x_2^2 + 2x_1x_3 = {}^t\boldsymbol{x}A\boldsymbol{x},\ \ \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ と表したとき、これを満たす実対称行列 A を求めよ．

(2) 固有方程式 $|A - \lambda E| = 0$ を解いて固有値 λ を求めよ．

(3) 固有値が小さい順に、それに対応する固有ベクトルを求めよ．

(4) 求めた固有ベクトルどうしの内積を計算せよ．

(5) 規格化された 3 つの直交する固有ベクトル $\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, \tilde{\boldsymbol{x}}_3$ を求めよ．

(6) 上記行列を対角化する直交行列 $P = (\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, \tilde{\boldsymbol{x}}_3)$を求めよ．

(7) 上記の直交行列を用いた直交変換 $P^{-1}AP$ により A が対角化されることを示せ．

(8) $f(\boldsymbol{x}) = 1$ が表す曲面について、(7)の直交変換に基づいて $\boldsymbol{y} = P^{-1}\boldsymbol{x}$ による座標変換を行い、$\boldsymbol{y}$ に基づいて曲面を描き、その種類を述べよ．

(9) この曲面は $\boldsymbol{y}$ で表したとき、その 2 つの軸と交わる．交点の座標を示せ．

試験科目	修士課程 （一般）
専門科目	

【問2】

x, yを非負の実数とし，下記の実数値関数

$$f(x,y)=\frac{1}{(3x+4y+1)^2}$$

について以下の問いに答えよ．ただし，答えだけでなく導出の過程を書くこと．

(1) hを実数とする．下記の極限に関する等式

$$\lim_{h\to 0}\frac{f(x+3h,y)-f(x,y-2h)}{h}=\frac{C}{(3x+4y+1)^3}$$

を満たす実数Cが存在する．このCを求めよ．

(2) hを実数とする．

$$f_X(x)=\lim_{h\to 0}\frac{f(x+h,1)-f(x,1)}{h},\quad f_Y(y)=\lim_{h\to 0}\frac{f(1,y+h)-f(1,y)}{h}$$

とおく．非負の実数a, bは$a^2+b^2=1$を満たす．$af_X(1)+bf_Y(1)$を最小化するa, bを求めよ．

(3) $g(x)=f(x,1)$とする．

$$S_n=\frac{1}{n}\sum_{k=1}^{n}g\left(\frac{k}{n}\right),\quad T_n=S_n-\frac{1}{n}\sum_{k=1}^{n}\frac{1}{\left(\frac{3k}{n}+5\right)\left\{\frac{3(k+1)}{n}+5\right\}}$$

とおく．$\displaystyle\lim_{n\to\infty}T_n$を求めよ．

(4) 下記の極限値

$$\lim_{n\to\infty}\sum_{k=1}^{n}\left\{\frac{1}{\frac{3k}{n}+5}-\frac{1}{\frac{3(k+1)}{n}+5}\right\}$$

を求めよ．

(5) $\displaystyle\lim_{n\to\infty}S_n$を求めよ．

2026 年度第 2 回法政大学大学院情報科学研究科入学者選抜試験
問題用紙

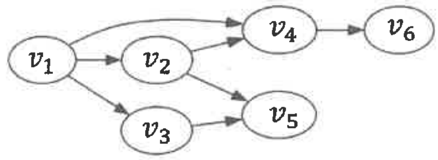
試 験 科 目	修士課程
専門科目	（一般）

【問 3】

(1) 有向グラフ$G = (V, E)$, $V = \{v_1, \ldots, v_m\}$の隣接行列とは、以下で定義される$m \times m$行列$A = (a_{ij})$を指す。

$$a_{ij} = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

ここで、a_{ij}は行列の第i行、第j列の要素を意味する。なお、グラフは有限であるとし、多重辺は含まないものとする。
問題：次の図で表されるグラフG_1の隣接行列を記せ。



(2) G_1の頂点v_1から、有向グラフにおける深さ優先探索を行ったとき、全ての頂点を訪問する順序を示せ。1,2,3,... のようにv_iの添え字の番号の列で示せ。複数通りの正解が考えられる場合には、そのうちの一つを示せばよい。

(3) 下記の「トポロジカルソートのアルゴリズム」は、有向非循環グラフGに対して、次の性質をもつように整列された頂点列を求めるアルゴリズムである。

性質：「Gの任意の 2 頂点a, bについてaからbへの有向辺がある場合、整列済みの頂点列内で、必ずaよりもbが列の後ろに配置される。」

問題：グラフG_1に対して、提示したトポロジカルソートのアルゴリズムを適用して得られる頂点列を、(2)と同様に添え字の番号の列で示せ。複数通りの正解が考えられる場合には、そのうちの一つを示せばよい。

トポロジカルソートのアルゴリズム:

入力：グラフG,　出力: トポロジカルソートされたGの頂点列

L ← 空リスト（結果の頂点列を格納する）

S ←Gにおける入力辺を持たない頂点の集合

WHILE S が空でない

　Sから一つの頂点nを選び、$S \leftarrow S \setminus \{n\}$とする($S$から$n$を除く)。

　Lの最後尾にnを追加する。

　Gにおけるnからの各出力辺$e = (n, m)$について

　　辺eをグラフGから除く。

　　IF 頂点mがGにおいて入力辺を持たない THEN

　　　$S \leftarrow S \cup \{m\}$ （S にmを追加する）

IF グラフの辺が空でない THEN エラー終了 ELSE Lを結果として出力する。

【問 3】の問題は次ページに続く

【問 3】（つづき）

(4) 下記の疑似コードにおける関数「トポロジカルソート(A)」は、上記に示した「トポロジカルソートのアルゴリズム」を、グラフGの隣接行列Aを用いて実装したものである。

<u>問題</u>：下記のプログラムで用いられている、下記の(a)(b)の関数を作成せよ。疑似コードで示すか、適当なプログラミング言語（手続き型言語または関数型言語）での実装を示せ。後者の場合には、記述言語(Python, Java, Haskell 等)を明記することとし、隣接行列のデータ表現は適宜データ型を各自で仮定してよい。

(a) 関数「**入力辺を持つかどうかを求める(A, i)**」：A は有向グラフの隣接行列、i は頂点v_iの添え字番号とし、頂点v_iが入力辺を持つ場合は真、そうでない場合は偽を返す。

(b) 関数「**入力辺を持たない頂点の集合を求める(A)**」：A は有向グラフの隣接行列とし、入力辺を持たない頂点の集合を返す。ただし、頂点の集合は、頂点記号v_iの添え字番号の集合として表す。

なお、下記の疑似コードは、配列の開始インデックスを 1（つまり 1-origin）とする記述言語を仮定しているが、回答に際して 0 を開始インデックスとする(つまり 0-origin の)プログラミング言語を記述言語として仮定する場合には、グラフ$G=(V,E)$における頂点集合Vは、$\{v_0, v_1, \dots, v_{m-1}\}$のように、頂点記号の添え字の番号が 0 から始まるものとして考えてよい。

```
// A の例： {{0, 1, 1},{0, 0, 1},{0, 0, 0}}
// 本疑似コードでは、配列は 1 を開始インデックスとする。(1-origin)
トポロジカルソート(A){
        N = G の頂点数; // すなわち A の行あるいは列の数 m
        L = {};
        S = 入力辺を持たない頂点の集合を求める(A); // (b)
        while (is_empty(S)==False){
          n = any(S);  S = remove(S, n);
          L = add_last(L, n);
          foreach i in {1,..., N} // 0-origin を仮定する場合は「foreach i in {0, ···, N-1}」に読み替える
                if (A[n][i] == 1){
                    A[n][i] = 0;
                    flag = 入力辺を持つかどうかを求める(A, i);  // (a)
                    if (flag == False)
                        S = add(S, i);
                }
        } // while のおわり
        if (グラフの辺が空ではない(A))
            エラー終了;
        else
            return L;
}
```