筆答専門試験科目（午前）　情報工学系

2025 大修

時間 ９：３０～１２：００

注　意　事　項

1．解答開始の指示があるまで，この問題冊子を開いてはならない．
2．各答案用紙の受験番号欄に受験番号を記入せよ．
3．次の１番～３番の３題すべてに解答せよ．
4．１枚の答案用紙に１題の回答を記入せよ．
5．各問題の解答は，それぞれの問題番号が記載された答案用紙に記入せよ．必要であれば，答案用紙の裏面にも記入して良い．
6．１枚の答案用紙に２題以上の解答を記入した場合は，それらの解答を無効とすることがある．
7．１題の解答を２枚以上の答案用紙に記入した場合は，その解答を無効とすることがある．
8．電子式卓上計算機等の使用は認めない．

1. 以下の問いに答えよ.

1) 次のそれぞれの実関数 f について，a) では 2 次の導関数を，b) では 2 次の全ての偏導関数を求めよ．ただし，$x, y \in (0, \infty)$ とし，α は正の定数とする.

$$\text{a)} \qquad f(x) = \left(\alpha + \sqrt{x}\right)^2$$

$$\text{b)} \qquad f(x, y) = \frac{\log_e(\alpha + x)}{\alpha + y}$$

2) $t \in (0, \pi)$ とするとき，以下の式によって与えられる xy 平面上の曲線と x 軸によって囲まれる領域の面積 A の値を求めよ.

$$\begin{cases} x(t) = 2t - \sin(2t) \\ y(t) = \dfrac{1 - \cos(2t)}{\sin(t)} \end{cases}$$

3) あるベクトル空間のベクトルの組 $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ が 1 次独立であるとき，以下のベクトルの組が 1 次独立であるかを a), b), c) のそれぞれについて答えよ.

a) $\mathbf{v}_1 - \mathbf{v}_2 + \mathbf{v}_3,\quad \mathbf{v}_1 + 2\mathbf{v}_2 - \mathbf{v}_3,\quad \mathbf{v}_2 + 4\mathbf{v}_3$

b) $\frac{1}{8}\mathbf{v}_1 - \frac{3}{16}\mathbf{v}_2 + \mathbf{v}_3,\quad \frac{3}{4}\mathbf{v}_1 + \frac{7}{8}\mathbf{v}_2 + \frac{1}{8}\mathbf{v}_3,\quad \frac{11}{16}\mathbf{v}_1 - \frac{3}{16}\mathbf{v}_2 - \frac{9}{8}\mathbf{v}_3,\quad \mathbf{v}_1 + \frac{1}{4}\mathbf{v}_2 - \frac{7}{16}\mathbf{v}_3$

c) $\frac{7}{8}\mathbf{v}_1 + \frac{3}{16}\mathbf{v}_2 - \frac{1}{2}\mathbf{v}_3,\quad -\frac{1}{8}\mathbf{v}_1 - \frac{1}{16}\mathbf{v}_2 + \frac{7}{16}\mathbf{v}_3$

4) $\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ は二変量正規分布に従う確率ベクトルであるとする．いま，平均ベクトルを $\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$，分散共分散行列を $\Sigma = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$ と与えるとき，点 $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ における X_1, X_2 の同時確率密度関数は次式で与えられる.

$$f_{X_1,X_2}(x_1, x_2) = \frac{1}{2\pi|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \boldsymbol{\mu}\right)^{\top} \Sigma^{-1} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \boldsymbol{\mu}\right)\right)$$

以下の問いに答えよ.

a) $X_1 + X_2$ の期待値と分散を求めよ.

b) X_1 で周辺化された X_2 の周辺確率密度関数 $f_{X_2}(x_2)$ を求めよ.

c) 条件 $X_2 = x_2$ の下での X_1 の確率密度関数 $f_{X_1|X_2}(x_1|x_2)$ を求めよ.

2. 次の問いに答えよ．ただし，解答では有限オートマトンの状態及び状態遷移は図 2.1 に例示するとおり表記することとし，状態のラベルは省略してもよい．有限オートマトンの状態遷移関数は部分関数として定義してもよい．また，空列を ε で表す．本問では導出過程を示す必要はない．

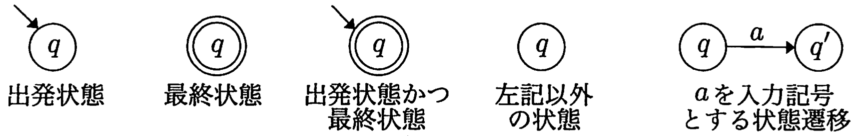

図 2.1: 有限オートマトンの状態及び状態遷移の表記

1) 正規文法 $G_1 = (N_1, T_1, P_1, S)$ について以下の問いに答えよ．ただし，$N_1 = \{S, X, Y, Z\}$ は非終端記号の集合，$T_1 = \{a, b\}$ は終端記号の集合，P_1 は次式で定義する生成規則の集合である．

$$
\begin{aligned}
P_1 = \{ & S \to aX, \\
& X \to aS \mid bY, \\
& Y \to b \mid bZ, \\
& Z \to bY \}
\end{aligned}
$$

また，S は出発記号である．G_1 が生成する言語を $L_1 = \mathcal{L}(G_1)$ とする．

a) 言語 L_1 の要素のうち長さ 5 の記号列をひとつ示せ．

b) 言語 L_1^* を生成する正規文法を $G_2 = (N_2, T_1, P_2, S')$ とする．ただし，$N_2 = \{S', S, X, Y, Z\}$ である．生成規則の集合 P_2 について，以下の空欄（ア）,（イ）,（ウ）を終端記号及び非終端記号からなる列で埋めよ．

$$
\begin{aligned}
P_2 = \{ & S' \to \varepsilon \mid \boxed{\text{（ア）}}, \\
& S \to aX, \\
& X \to aS \mid bY, \\
& Y \to b \mid bZ \mid \boxed{\text{（イ）}}, \\
& Z \to \boxed{\text{（ウ）}} \}
\end{aligned}
$$

c) 言語 L_1L_1 を生成する正規文法 G_3 を示せ．ただし，非終端記号の集合の要素数は 8 以下とする．

d) 3 つの言語 L_1，L_1^*，L_1L_1 のうち，ε を要素として含むものをすべて挙げよ．該当する言語がない場合は「なし」と解答せよ．

（次ページにつづく）

（前ページのつづき）

2) 決定性有限オートマトン $M_1 = (K, T, t, q_0, F)$ について以下の問いに答えよ．ただし，$K = \{q_0, q_1, q_2, q_3\}$ は状態集合，$T = \{0, 1\}$ は入力アルファベット，t は図 2.2 の状態遷移図で表される状態遷移関数であり，例えば，$t(q_0, 1) = q_1$ である．また，q_0 は出発状態，$F = \{q_1\}$ は最終状態の集合である．M_1 が受理する言語を L_2 とする．

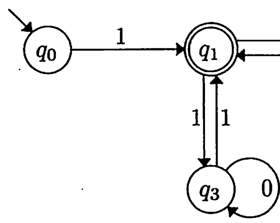

図 2.2: M_1 の状態遷移図

a) $t(q_1, 0)$，$t(q_1, 1)$，$t(q_2, 0)$，$t(q_2, 1)$ の値をそれぞれ示せ．

b) 言語 L_2 の要素のうち長さ 4 以下の記号列をすべて示せ．

c) 言語 L_2 の補集合 $T^* \setminus L_2$ を受理する決定性有限オートマトン M_2 の状態遷移図を示せ．ただし，M_2 の状態数は 5 以下とする．

d) 文法 $G_4 = (N_4, T, P_4, S)$ が生成する言語を $\mathcal{L}(G_4)$ とする．ただし，$N_4 = \{S, X\}$ であり，P_4 は次式で定義する．

$$\begin{aligned} P_4 = \{ & S \to 1S \mid 0X, \\ & X \to 1S \mid 0X \mid \varepsilon\} \end{aligned}$$

図 2.3 の状態遷移図で表される決定性有限オートマトン M_3 が言語 $L_2 \cap \mathcal{L}(G_4)$ を受理するように，空欄（ア）から（ケ）をそれぞれ 1 つの入力記号で埋めよ．

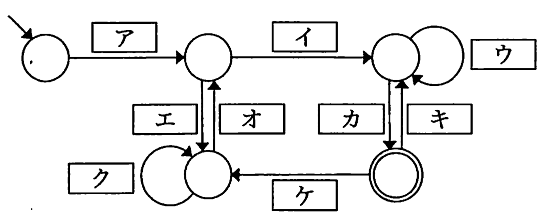

図 2.3: M_3 の状態遷移図

（次ページにつづく）

（前ページのつづき）

3) 以下の正規表現で与えられるアルファベット $\Sigma = \{a, b\}$ 上の言語を受理する，状態数最小の決定性有限オートマトンの状態遷移図を示せ．

$$(ab^*a + bba^*)ab^*$$

4) 以下の (i) から (v) に示す，アルファベット $\Sigma = \{a, b, c\}$ 上の言語が属する言語の族について適切な記述を，下の選択肢（ア），（イ），（ウ）の中からそれぞれ 1 つずつ選べ．

(i) $\left\{a^s bc \,\middle|\, s = 2^i, i \in \mathbb{Z}^+\right\}$

(ii) $\left\{a^s b^s c^t \,\middle|\, s, t \in \mathbb{Z}^+\right\}$

(iii) $\left\{a^s b^t c^{2u} \,\middle|\, s, t, u \in \mathbb{Z}^+\right\}$

(iv) $\left\{a^s b^s c^t \,\middle|\, s, t \in \mathbb{Z}^+\right\} \cap \left\{a^s b^t c^{2s} \,\middle|\, s, t \in \mathbb{Z}^+\right\}$

(v) $\left\{a^s b^{2t} c^{s+t} \,\middle|\, s, t \in \mathbb{Z}^+\right\}$

ただし，$\mathbb{Z}^+$ は正整数の集合である．

（選択肢）

（ア）正規言語である

（イ）文脈自由言語であり，かつ正規言語ではない

（ウ）文脈自由言語ではない

3.

以下の問いに答えよ．ただし，正しい解の組合せが複数ある場合はいずれかひとつの組を書け．非負の整数 W と H に対して，スタート地点 $(0,0)$ からゴール地点 (W,H) まで整数座標の点 (x,y) のみを通過して移動する経路の数を数える C 言語のプログラムを考える．ただし，$0 \leq x \leq W$ かつ $0 \leq y \leq H$ であり，点から点への移動は上下左右のみとし，斜めの移動はできないものとする．本問では移動した線分の長さの総和を経路長とする．なお，スタート地点とゴール地点が同一点である場合は，経路長 0 の経路が 1 つあるとみなす．また，プログラム実行中にはオーバーフローはしないものとする．

まず，以下の問い 1)〜6) では，最短経路の数を考える．ここで，最短経路とは経路長が最短となる経路のことを指す．たとえば，$W=2$，$H=2$ のときの最短経路の数は 6 である．

1) $W=6$，$H=5$ のときの最短経路の数を求めよ．
2) ソースコード 3.1 は引数 w に W を，h に H を渡して実行することで最短経路の数を返す関数の例である．空欄 $\boxed{\text{A}}$〜$\boxed{\text{C}}$ に入るものを次ページの表 3.1 の選択肢群から選び，番号で答えよ．同じ選択肢を複数回選んでもよい．

```
int count_shortest_path(int w, int h) {
    if (w == 0 || h == 0)
        return [ A ];
    return count_shortest_path(w, [ B ]) + count_shortest_path([ C ], h);
}
```

ソースコード 3.1

3) ソースコード 3.2 は引数 w に W を，h に H を渡して実行することで最短経路の数を返す関数の別例である．ただし，グローバル変数 a は int 型の要素からなる 2 次元配列であり，要素数は両次元ともに十分大きいものとする．空欄 $\boxed{\text{あ}}$〜$\boxed{\text{き}}$ に入るものを次ページの表 3.1 の選択肢群から選び，番号で答えよ．同じ選択肢を複数回選んでもよい．

```
int count_shortest_path(int w, int h) {
    for (int i = 0; i <= h; i++)
        a[i][0] = [ あ ];
    for (int i = 0; i <= w; i++)
        a[0][i] = [ い ];
    for (int i = 1; i <= h; i++)
        for (int j = 1; j <= w; j++)
            a[i][j] = a[i][[ う ]] + a[[ え ]][[ お ]];
    return a[[ か ]][[ き ]];
}
```

ソースコード 3.2

（次ページにつづく）

表 3.1

選択肢群							
① `1`	② `0`	③ `w`	④ `w-1`	⑤ `w+1`	⑥ `w+h`	⑦ `h`	⑧ `h-1`
⑨ `h+1`	⑩ `i`	⑪ `i-1`	⑫ `i+j`	⑬ `j`	⑭ `j-1`	⑮ `j-i`	

4) $W = H = N$ の場合にソースコード 3.2 で定義された関数による a の要素の代入回数を $f(N)$ とおく．$f(N) = O(g(N))$ をみたす $g(N)$ のうち，N を十分に大きくしたときの N に対する増加量が最も小さいものを以下の選択肢群から選び，番号で答えよ．

選択肢群				
① 1	② $\log N$	③ N	④ $N \log N$	⑤ $N^2 \log N$
⑥ N^2	⑦ 2^N	⑧ $N!$	⑨ $2^N \log N$	⑩ $N! \log N$

5) ソースコード 3.2 を用いた count_shortest_path(17, 13) の実行が完了した直後の配列 a の要素の一部を下記に示す．$W = 17$，$H = 13$ のとき，点 $(8, 7)$ が通行止めとなり通れなくなった場合の最短経路の数を求めよ．

a[6][7]=1716,	a[6][8]=3003,	a[6][9]=5005,	a[7][7]=3432,
a[7][8]=6435,	a[7][9]=11440,	a[8][7]=6435,	a[8][8]=12870,
a[8][9]=24310,	a[13][17]=119759850		

6) 前問 5) と同様に $W = 17$，$H = 13$ かつ点 $(8, 7)$ が通行止めの場合の最短経路の数を求める問題を考える．通行止めの点を通る経路の数が 0 になると考え，ソースコード 3.1 の 3 行目の直後に下記の 2 行を挿入した．変更後のソースコードで count_shortest_path(17, 13) を実行した結果が問題の答えとして正しいか否かを答えよ．正しくない場合は，その理由も書け．

```
    if (w == 8 && h == 7)
        return 0;
```

（次ページにつづく）

（前ページのつづき）

次に，以下の問い 7)～9) では，最短経路だけではなく，経路長が非最短の場合も含むすべての経路を考える．ただし，ひとつの経路においては一度通った点は二度と通れないものとする．たとえば，$W = 2$，$H = 2$ のときの経路の数は 12 である．

$W = H = N$ の場合の経路の数を返す関数 count_path を実装する．この関数を用いて $N = 3$ の場合の経路の数を求める main 関数の例をソースコード 3.3 に，関数 count_path の例を次ページのソースコード 3.4 に示す．一度通った点を記録するために，図 3.1 に示すように，構造体 Node を要素とする連結リストを用いる．関数 is_visited は，ある点を表す Node 値を引数とし，その点を以前に通過したか否かを返す．

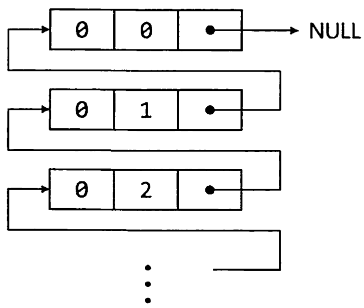

図 3.1

```c
int main() {
    struct Node n;
    n.x = 0;  n.y = 0;  n.p = NULL;
    printf("%d\n", count_path(n, 3));
    return 0;
}
```

ソースコード 3.3

7) ソースコード 3.4 の空欄 $\boxed{\text{ア}}$～$\boxed{\text{ノ}}$ に入るものを以下の選択肢群から選び，番号で答えよ．同じ選択肢を複数回選んでもよい．

選択肢群

① 1	② 0	③ N	④ N-1	⑤ n
⑥ *n	⑦ &n	⑧ n.x	⑨ n.y	⑩ n.p
⑪ n->x	⑫ n->y	⑬ n->p	⑭ n->p->p	⑮ n_curr
⑯ *n_curr	⑰ &n_curr	⑱ n_curr.x	⑲ n_curr.y	⑳ n_curr.p
㉑ n_curr->x	㉒ n_curr->y	㉓ n_curr->p	㉔ n_new	㉕ *n_new
㉖ &n_new	㉗ n_new.p	㉘ n_new->p	㉙ NULL	㉚ &&
㉛ ||	㉜ int	㉝ void	㉞ struct Node	㉟ struct Node*

（次ページにつづく）

（前ページのつづき）

```
#include <stdio.h>

struct Node {
    int x;
    int y;
    [ア] p;
};

int is_visited(struct Node n) {
    [イ] n_curr = [ウ];
    while ([エ] != NULL) {
        n_curr = [オ];
        if ([カ] == n.x [キ] [ク] == n.y)
            return 1;
    }
    return 0;
}

int count_path(struct Node n, int N) {
    int dx[] = {0, 0, -1, 1};
    int dy[] = {-1, 1, 0, 0};
    int res = 0;
    if (is_visited([ケ]))
        return 0;
    if (n.x == [コ] [サ] n.y == [シ])
        return [ス];
    for (int i = 0; i < 4; i++) {
        int x = [セ] + dx[i];
        int y = [ソ] + dy[i];
        struct Node n_new;
        if (x < [タ] [チ] y < [ツ] [テ] x > [ト] [ナ] y > [ニ])
            continue;
        n_new.x = x;  n_new.y = y;  n_new.p = [ヌ];
        res += count_path([ネ], [ノ]);
    }
    return res;
}
```

ソースコード 3.4

（次ページにつづく）

（前ページのつづき）

8) 下記の 3 行を，ソースコード 3.4 の 32 行目の直後に挿入した場合を考える．この場合にソースコード 3.3 の main 関数を実行したとき，挿入された printf 関数による印字結果を書け．ただし，int 型のグローバル変数 t は 0 で初期化してあるものとする．

```
        t++;
        if (t < 5)
            printf("%d %d\n", x, y);
```

9) 前問 8) と同じ 3 行を，ソースコード 3.4 の（32 行目ではなく）34 行目の直後に挿入した場合を考える．この場合にソースコード 3.3 の main 関数を実行したとき，挿入された printf 関数による印字結果を書け．ただし，int 型のグローバル変数 t は 0 で初期化してあるものとする．