専門科目 (午前)

29　大修

数理・計算科学

Mathematical and Computing Science

時間 午前 9 時 30 分 – 午後 1 時

Time 9:30AM – 1:00PM

注意事項

1. 問 A, 問 B, 問 C より 2 問を選択し解答せよ.
2. 問 1〜問 9 より 3 問を選択し解答せよ.
3. 要求された問題数を超えて解答した場合は採点されない可能性がある.
4. すべての解答用紙に必ず問題番号および受験番号を記入せよ.
5. 解答は 1 問ごとに 1 枚の解答用紙に記入せよ.
6. 解答用紙の裏面を使用しても構わないが，その場合は表に「裏面へ続く」等の表示を書いておくこと.

Instruction

1. Solve 2 problems out of Problems A, B, and C.
2. Solve 3 problems out of Problems 1 to 9.
3. Note that if you solved more problems than specified above, problems you solved might not be scored.
4. Write the problem number and your examinee number in the designated place of each answer sheet.
5. Use one answer sheet per problem.
6. You may use the other side of the answer sheet, but in that case you should indicate that, for example, by writing “continue to the other side.”

問 A

n 次実対称行列 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ と n 次元実ベクトル $\boldsymbol{b} \in \mathbb{R}^n$ が与えられているとし，関数 $f:\mathbb{R}^n \to \mathbb{R}$ を $f(\boldsymbol{x}) := \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}$ と定義する．ただし，$\boldsymbol{x}^\top$ はベクトル $\boldsymbol{x}$ の転置を表すとする．

(1) $\boldsymbol{A}$ が正定値行列，つまり任意の $\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{x} \neq \boldsymbol{0}$ に対して $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} > 0$ であれば，$f(\boldsymbol{x})$ は最小値をとり，その値を達成するような点（今後，そのような点を最小点と呼ぶ）が存在し，一意であることを示せ．

(2) $\boldsymbol{A}$ が半正定値行列，つまり任意の $\boldsymbol{x} \in \mathbb{R}^n$ に対して $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} \geq 0$ であり，$\boldsymbol{A}$ の階数（ランク）を k $(1 \leq k < n)$ とする．$f(\boldsymbol{x})$ の最小点からなる集合 S から任意の $\boldsymbol{x}^* \in S$ をとってきたとすると，$S_{\boldsymbol{x}^*} = \{\boldsymbol{s} - \boldsymbol{x}^* | \boldsymbol{s} \in S\}$ が $\mathbb{R}^n$ の線形部分空間になることを示せ．

(3) $S_{\boldsymbol{x}^*}$ の次元を n と k を用いて求めよ．

Problem A

Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be an n-order real symmetric matrix, $\boldsymbol{b} \in \mathbb{R}^n$ be an n-dimensional real vector, and let us define the function $f : \mathbb{R}^n \to \mathbb{R}$ as $f(\boldsymbol{x}) := \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}$. Here, $\boldsymbol{x}^\top$ means the transpose of the vector $\boldsymbol{x}$.

(1) Assume $\boldsymbol{A}$ is a positive definite matrix (that is, $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} > 0$, for all $\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{x} \neq \boldsymbol{0}$). Show that there is a point where $f(\boldsymbol{x})$ attains its minimum (we call this/these point(s) the minimal point(s), hereafter), and it is unique.

(2) Assume $\boldsymbol{A}$ is positive semidefinite (that is, $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} \geq 0$, for all $\boldsymbol{x} \in \mathbb{R}^n$), and also that the rank of $\boldsymbol{A}$ is k $(1 \leq k < n)$. Define S as the set of all the minimal points of $f(\boldsymbol{x})$, and let an arbitrary $\boldsymbol{x}^* \in S$. Show that $S_{\boldsymbol{x}^*} = \{\boldsymbol{s} - \boldsymbol{x}^* | \boldsymbol{s} \in S\}$ is a linear subspace of $\mathbb{R}^n$.

(3) Determine the dimension of $S_{\boldsymbol{x}^*}$ in terms of n and k.

問 B

$[0, \infty)$ 上の実数値連続関数 f が，ある実数 A に対し $\lim_{x \to \infty} f(x) = A$ を満たすと仮定する．以下の問に答えよ．

(1) $[0, \infty)$ 上, f は有界であることを示せ．

(2) 極限

$$\lim_{a \to \infty} \frac{1}{a^2} \int_0^a f(x) dx$$

を求めよ．

(3) 極限

$$\lim_{a \to \infty} \frac{1}{a} \int_0^a f(x) dx$$

を求めよ．

Problem B

Let f be a real-valued continuous function over $[0, \infty)$ which converges to a certain real value A as $x \to \infty$, i.e., $\lim_{x \to \infty} f(x) = A$. Answer the following questions.

(1) Show that f is bounded over $[0, \infty)$.

(2) Find the following limit

$$\lim_{a \to \infty} \frac{1}{a^2} \int_0^a f(x) dx.$$

(3) Find the following limit

$$\lim_{a \to \infty} \frac{1}{a} \int_0^a f(x) dx.$$

問 C

$\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ とし，これらの文字から成る有限列全体を Σ^* と表記する．以下では $\alpha, \beta, \ldots$ 等は Σ^* の要素を表す．列 $\omega \in \Sigma^*$ が与えられたとき Σ^* 上の二項関係 $\overset{(\omega)}{\mapsto}$ と $\overset{(\omega)}{\rightsquigarrow}$ を次のように定義する．

$\alpha \overset{(\omega)}{\mapsto} \beta \iff$ α 中に出現するすべての a を ω に置き換えると β になる．

$\alpha \overset{(\omega)}{\rightsquigarrow} \beta \iff$ α 中に出現する a のいくつか（0 個でもよい）を ω に置き換えると β になる．

たとえば次が成り立つ．

$\mathsf{abaabb} \overset{(\mathsf{cac})}{\mapsto} \mathsf{cacbcaccacbb}$

$\mathsf{abaabb} \overset{(\mathsf{cac})}{\rightsquigarrow} \mathsf{cacbcaccacbb}$ $\quad$ $\mathsf{abaabb} \overset{(\mathsf{cac})}{\rightsquigarrow} \mathsf{abcacabb}$ $\quad$ $\mathsf{abaabb} \overset{(\mathsf{cac})}{\rightsquigarrow} \mathsf{abaabb}$

$\mathsf{abaabb} \overset{(\mathsf{a})}{\mapsto} \mathsf{abaabb}$ $\quad$ $\mathsf{abaabb} \overset{(\varepsilon)}{\mapsto} \mathsf{bbb}$ $\quad$ $\mathsf{abaabb} \overset{(\varepsilon)}{\rightsquigarrow} \mathsf{abbb}$

$\mathsf{ccc} \overset{(\mathsf{abc})}{\mapsto} \mathsf{ccc}$

ただし ε は空列である．さらに Σ^* 上の二項関係 $\mapsto$ と $\rightsquigarrow$ を次のように定義する．

$\alpha \mapsto \beta \iff$ ある $\omega \in \Sigma^*$ が存在して $\alpha \overset{(\omega)}{\mapsto} \beta$.

$\alpha \rightsquigarrow \beta \iff$ ある $\omega \in \Sigma^*$ が存在して $\alpha \overset{(\omega)}{\rightsquigarrow} \beta$.

(1) $\alpha = \mathsf{abac}$, $\beta = \mathsf{caabcaac}$, $\gamma = \mathsf{cbababcbabac}$ とすると $\alpha \overset{(\mathsf{caa})}{\mapsto} \beta \overset{(\mathsf{ba})}{\mapsto} \gamma$ である．このとき $\alpha \overset{(\omega)}{\mapsto} \gamma$ となる ω を書け．

(2) $\mapsto$ が推移的であることを示せ．

(3) $\rightsquigarrow$ が推移的でないことを示せ．

(4) $\mapsto$ が反対称的であることを示せ．

(5) $\rightsquigarrow$ が反対称的でないことを示せ．

(注) 集合 X 上の二項関係 R が推移的であるとは，任意の $x, y, z \in X$ に対して

$(xRy$ かつ $yRz)$ ならば xRz

が成り立つことである．また R が反対称的であるとは，任意の $x, y \in X$ に対して

$(xRy$ かつ $yRx)$ ならば $x = y$

が成り立つことである．

Problem C

Define $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$. Σ^* represents the set of finite strings over Σ. In the following, $\alpha, \beta, \ldots$ denote elements of Σ^*. Given a string $\omega \in \Sigma^*$, we define two binary relations $\overset{(\omega)}{\mapsto}$ and $\overset{(\omega)}{\leadsto}$ on Σ^* as follows.

$$\alpha \overset{(\omega)}{\mapsto} \beta \iff \beta \text{ is obtained from } \alpha \text{ by replacing all the occurrences of } \mathsf{a} \text{ by } \omega.$$

$$\alpha \overset{(\omega)}{\leadsto} \beta \iff \beta \text{ is obtained from } \alpha \text{ by replacing some (possibly zero) occurrences of } \mathsf{a} \text{ by } \omega.$$

For example, we have

$\mathsf{abaabb} \overset{(\mathsf{cac})}{\mapsto} \mathsf{cacbcaccacbb}$

$\mathsf{abaabb} \overset{(\mathsf{cac})}{\leadsto} \mathsf{cacbcaccacbb}$ $\qquad$ $\mathsf{abaabb} \overset{(\mathsf{cac})}{\leadsto} \mathsf{abcacabb}$ $\qquad$ $\mathsf{abaabb} \overset{(\mathsf{cac})}{\leadsto} \mathsf{abaabb}$

$\mathsf{abaabb} \overset{(\mathsf{a})}{\mapsto} \mathsf{abaabb}$ $\qquad$ $\mathsf{abaabb} \overset{(\varepsilon)}{\mapsto} \mathsf{bbb}$ $\qquad$ $\mathsf{abaabb} \overset{(\varepsilon)}{\leadsto} \mathsf{abbb}$

$\mathsf{ccc} \overset{(\mathsf{abc})}{\mapsto} \mathsf{ccc}$

where ε denotes the empty string. Moreover we define binary relations $\mapsto$ and $\leadsto$ on Σ^* as follows.

$$\alpha \mapsto \beta \iff \alpha \overset{(\omega)}{\mapsto} \beta \text{ for some } \omega \in \Sigma^*.$$

$$\alpha \leadsto \beta \iff \alpha \overset{(\omega)}{\leadsto} \beta \text{ for some } \omega \in \Sigma^*.$$

(1) Let $\alpha = \mathsf{abac}$, $\beta = \mathsf{caabcaac}$, and $\gamma = \mathsf{cbababcbabac}$. We have $\alpha \overset{(\mathsf{caa})}{\mapsto} \beta \overset{(\mathsf{ba})}{\mapsto} \gamma$. Find a string ω such that $\alpha \overset{(\omega)}{\mapsto} \gamma$.

(2) Show that the relation $\mapsto$ is transitive.

(3) Show that the relation $\leadsto$ is not transitive.

(4) Show that the relation $\mapsto$ is antisymmetric.

(5) Show that the relation $\leadsto$ is not antisymmetric.

(Note) A binary relation R on a set X is transitive if xRy and yRz imply xRz, for all $x, y, z \in X$. R is antisymmetric if xRy and yRx imply $x = y$, for all $x, y \in X$.

問 1

2 元の有限体 $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z} = \{\bar{0}, \bar{1}\}$ について，群

$$\begin{aligned} G &= GL_2(\mathbb{F}_2) \\ &= \left\{ g = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mid g \text{ は正則, } a, b, c, d \in \mathbb{F}_2 \right\} \end{aligned}$$

を考える．群 G の積は行列の積であり，単位元は単位行列 $\begin{pmatrix} \bar{1} & \bar{0} \\ \bar{0} & \bar{1} \end{pmatrix}$ である．

(1) 群 G の元 $x = \begin{pmatrix} \bar{1} & \bar{1} \\ \bar{0} & \bar{1} \end{pmatrix}$ と $y = \begin{pmatrix} \bar{0} & \bar{1} \\ \bar{1} & \bar{1} \end{pmatrix}$ について，位数をそれぞれ求めよ．

(2) 群 G の位数を求めよ．

(3) 群 G の部分群をすべて求めよ．

Problem 1

For the finite field $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z} = \{\bar{0}, \bar{1}\}$ with two elements, consider the group

$$\begin{aligned} G &= GL_2(\mathbb{F}_2) \\ &= \left\{ g = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mid g \text{ is non-singular}, \ a, b, c, d \in \mathbb{F}_2 \right\}. \end{aligned}$$

Remark that the multiplication of the group G is the matrix multiplication, and the identity element is the identity matrix $\begin{pmatrix} \bar{1} & \bar{0} \\ \bar{0} & \bar{1} \end{pmatrix}$.

(1) Find the order of the elements $x = \begin{pmatrix} \bar{1} & \bar{1} \\ \bar{0} & \bar{1} \end{pmatrix}$ and $y = \begin{pmatrix} \bar{0} & \bar{1} \\ \bar{1} & \bar{1} \end{pmatrix}$ in the group G, respectively.

(2) Find the order of the group G.

(3) Find all subgroups of the group G.

問 2

有限集合 $S = \{a, b, c, d\}$ について，部分集合の族

$$\mathcal{O} = \{\emptyset, \{a\}, \{c\}, \{a, c\}, \{b, c, d\}, S\}$$

を考える．

(1) $\mathcal{O}$ が S の開集合系であることを示せ．

(2) 位相空間 $(S, \mathcal{O})$ の部分集合 $\{b\}$ の閉包を求めよ．

(3) 位相空間 $(S, \mathcal{O})$ はハウスドルフ空間になるか否かを理由をつけて述べよ．

Problem 2

For the finite set $S = \{a, b, c, d\}$, consider the family of subsets

$$\mathcal{O} = \{\emptyset, \{a\}, \{c\}, \{a, c\}, \{b, c, d\}, S\}.$$

(1) Show that $\mathcal{O}$ is a topology on S.

(2) Find the closure of the subset $\{b\}$ in the topological space $(S, \mathcal{O})$.

(3) Determine whether the topological space $(S, \mathcal{O})$ is a Hausdorff space or not.

問 3

$[0,1]$ 上の連続関数全体のなす線形空間 $C[0,1]$ の元 u に対して，

$$\|u\| = \max\{\,|u(x)|\,; 0 \leq x \leq 1\}$$

とおく．以下の問に答えよ．

(1) $\|\cdot\|$ が $C[0,1]$ のノルムとなることを示せ．

以下，このノルムの導入された線形空間 $C[0,1]$ を X と表す．

(2) P を多項式全体のなす集合とする．P は X の部分空間であることを示せ．ここで，X の部分空間とは，X の部分集合で，線形空間となるものを意味する．

(3) 自然数 n に対して，$u_n \in P$ を

$$u_n(x) = 1 + \frac{x}{2} + \left(\frac{x}{2}\right)^2 + \left(\frac{x}{2}\right)^3 + \cdots + \left(\frac{x}{2}\right)^{n-1}$$

と定義する．$n \to \infty$ のとき，u_n は X で収束することを示せ．

(4) $n \to \infty$ のとき，u_n は P で収束しないことを示せ．

Problem 3

Let $C[0,1]$ be the linear space of continuous functions over $[0,1]$. Let

$$\|u\| = \max\{\, |u(x)|\,; 0 \leq x \leq 1\}$$

for $u \in C[0,1]$. Answer the following questions.

(1) Show that $\| \cdot \|$ is a norm of $C[0,1]$.

Hereafter, X denotes the linear space $C[0,1]$ with this norm.

(2) Let P be the set of polynomials. Show that P is a subspace of X. Here a subspace of X means a linear space which is a subset of X.

(3) For each natural number n, let $u_n \in P$ be defined by

$$u_n(x) = 1 + \frac{x}{2} + \left(\frac{x}{2}\right)^2 + \left(\frac{x}{2}\right)^3 + \cdots + \left(\frac{x}{2}\right)^{n-1}.$$

Show that u_n converges in X as $n \to \infty$.

(4) Show that u_n does not converge in P as $n \to \infty$.

問 4

次の線形計画問題 $\mathcal{P}$ を考える：

$$\mathcal{P}\ :\ \begin{array}{lcrcl} \text{最大化} & : & -2x_1+5x_2-4x_3 & & \\ \text{制約} & : & 3x_1+3x_2-5x_3 & \leq & 7 \\ & & 7x_1-3x_2+7x_3 & \leq & 4 \\ & & x_1,x_2,x_3 & \geq & 0. \end{array}$$

以下の問に答えよ.

(1) $\mathcal{P}$ の最適解 (x_1^*, x_2^*, x_3^*) をシンプレックス法で求めよ.

(2) $\mathcal{P}$ の双対問題 $\mathcal{D}$ を書き下せ.

(3) $\mathcal{D}$ の実行可能集合を図示し，その唯一の端点を求めよ.

Problem 4

Let $\mathcal{P}$ be the following linear programming problem

$$\mathcal{P} \quad : \quad \begin{array}{lcrcl} \text{maximize} & : & -2x_1 + 5x_2 - 4x_3 & & \\ \text{subject to} & : & 3x_1 + 3x_2 - 5x_3 & \leq & 7 \\ & & 7x_1 - 3x_2 + 7x_3 & \leq & 4 \\ & & x_1, x_2, x_3 & \geq & 0. \end{array}$$

Answer the following questions.

(1) Obtain an optimal solution (x_1^*, x_2^*, x_3^*) of $\mathcal{P}$ using the simplex method.

(2) Write the dual problem $\mathcal{D}$ of $\mathcal{P}$.

(3) Draw the feasible set of $\mathcal{D}$ and compute its unique vertex.

問 5

以下の問に答えよ．

(1) 確率 $p \in (0,1)$ で表の出るコイン投げを考える．表の出るまで投げた回数の期待値と分散を p を用いて表せ．

(2) ある袋の中に N 種類の玉が一つずつ入っているものとする．ここで N は 2 以上の整数とする．袋の中から無作為に玉を一つ取り出し，玉の種類を記録して袋の中に戻すことを「試行」と呼ぶとき，全ての種類の玉を記録するために何回の試行が必要か，という問題を考えよう．

各試行は独立で，取り出される玉の種類は一様分布に従うとする．さらに，k 種類記録するまでに要した試行回数を S_k と記す．このとき，以下の問 (i),(ii) に答えよ．

(i) $S_{k+1} - S_k$ の期待値と分散を N と k を用いて表せ．ただし，$S_{k+1} - S_k$ が残り $N - k$ 種類の内の一つを取り出すまでに繰り返した試行の数であることに注意せよ．

(ii) S_N の期待値と分散をそれぞれ μ, σ^2 とおくとき，

$$\mu = N \sum_{i=1}^{N} \frac{1}{i}, \quad \sigma^2 = N \sum_{i=1}^{N-1} \frac{N-i}{i^2}$$

であることを証明せよ．

Problem 5

Answer the following questions:

(1) We consider the coin tosses where the probability of a head coming up is $p \in (0, 1)$. Express, in terms of p, the expectation and the variance of the number of tosses needed to see the first head.

(2) Suppose that there is a bag with N-types of balls, each type consisting of exactly one ball. Here, N is an integer greater than or equal to 2. We call as a *trial* the following set of procedures of sampling a ball at random from the bag, recording the type of the sampled ball, and then putting it back in the bag. Then let us consider the problem of finding the number of trials needed to record all types of the balls.

We assume that each trial is independent of the others, and that the type of sampled balls follows the uniform distribution. Moreover, we denote by S_k the number of trials needed to record k-types of the balls. Then answer the following questions (i) and (ii):

(i) Express the expectation and the variance of $S_{k+1} - S_k$ in terms of N and k. Here, you can use the fact that $S_{k+1} - S_k$ is the number of trials needed to get one of the last $(N - k)$-types of the balls.

(ii) Show that the expectation μ and the variance σ^2 of S_N are given by

$$\mu = N \sum_{i=1}^{N} \frac{1}{i}, \quad \sigma^2 = N \sum_{i=1}^{N-1} \frac{N-i}{i^2}.$$

問 6

２つの店 A, B があるとする．各店の毎日の来客人数はそれぞれ独立なポアソン分布 $\mathrm{Po}(\lambda)$，$\mathrm{Po}(\nu)$ に従うとする．１日あたりの店 A と店 B の来客人数をそれぞれ X, Y と書き，n 日間の来客人数のデータ $(x_i, y_i)_{i=1}^n = ((x_1, y_1), \ldots, (x_n, y_n))$ が得られているとする．以下の問に答えよ．なお，導出の過程も記述すること．
ただし，パラメータ $\lambda > 0$ のポアソン分布 $\mathrm{Po}(\lambda)$ に従う確率変数 X の確率質量関数は

$$\mathrm{P}(X = k; \lambda) = \frac{\lambda^k}{k!} e^{-\lambda} \ (k = 0, 1, 2, \ldots) \qquad (*)$$

で与えられる．

(1) ポアソン分布 $(*)$ が実際に確率分布になっていることを示せ．すなわち，$\sum_{k=0}^{\infty} \mathrm{P}(X = k; \lambda) = 1$ を示せ．

(2) 観測データ $(x_i, y_i)_{i=1}^n$ から λ と ν の最尤推定量を求めよ．ただし，$\sum_{i=1}^n x_i \neq 0$ かつ $\sum_{i=1}^n y_i \neq 0$ とする．

(3) ある日の合計来客人数 $N = X + Y$ がわかっているとき，X の条件付き分布を求めよ．

(4) 実は Y は X とは独立ではなく，X が与えられたときの Y の条件付き分布が未知パラメータ $a > 0$ を用いて

$$\mathrm{P}(Y = \ell | X = k; a) = \frac{f_a(k)^\ell}{\ell!} \exp(-f_a(k)),$$
$$f_a(k) = a(1 + k)$$

なるモデルで記述できるとする．観測データ $(x_i, y_i)_{i=1}^n$ から a の最尤推定量を求めよ．ただし，$\sum_{i=1}^n y_i \neq 0$ とする．

Problem 6

Suppose that there are two stores A and B. The numbers of customers arriving at stores A and B in each day obey independent Poisson distributions $\mathrm{Po}(\lambda)$ and $\mathrm{Po}(\nu)$, respectively. We write the numbers of customers arriving at the stores A and B in a day as X and Y, respectively. Assume that we observed the data $(x_i, y_i)_{i=1}^n = ((x_1, y_1), \ldots, (x_n, y_n))$ of the numbers of customers for n days. Answer the following questions.
Here, the probability mass function of X obeying the Poisson distribution $\mathrm{Po}(\lambda)$ with the parameter $\lambda > 0$ is given by

$$\mathrm{P}(X = k; \lambda) = \frac{\lambda^k}{k!} e^{-\lambda} \ (k = 0, 1, 2, \ldots). \qquad (*)$$

(1) Show that the Poisson distribution $(*)$ is actually a probability distribution. That is, show that $\sum_{k=0}^{\infty} \mathrm{P}(X = k; \lambda) = 1$.

(2) Suppose that $\sum_{i=1}^n x_i \neq 0$ and $\sum_{i=1}^n y_i \neq 0$. Derive the maximum likelihood estimators of λ and ν from the observed data $(x_i, y_i)_{i=1}^n$.

(3) Given the total number $N = X + Y$ of customers, derive the conditional distribution of X.

(4) Suppose that Y is not independent of X, and that the conditional distribution of Y given X is expressed by the following model:

$$\mathrm{P}(Y = \ell | X = k; a) = \frac{f_a(k)^\ell}{\ell!} \exp(-f_a(k)),$$
$$f_a(k) = a(1 + k)$$

where $a > 0$ is an unknown parameter. Suppose that $\sum_{i=1}^n y_i \neq 0$. Derive the maximum likelihood estimator of a from the observed data $(x_i, y_i)_{i=1}^n$.

問 7

アルファベット

$$\Sigma = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$$

上の言語に関して以下の問に答えよ．

(1) 言語

$$A = \{ w \mid w \text{ の上段の文字列は回文} \}$$

とする．回文とは前から読んでも後ろから読んでも同じになる文字列である．また，空列 ε も回文とみなす．たとえば，

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in A$$

である．ポンピング補題を用いて A が正規でないことを示せ．

(2) 言語 A を生成する文脈自由文法を与えよ．

(3) ここでは，アルファベット Σ 上の文字列の各行を 2 進数とみなす．ただし，左端が最上位ビット (MSB) とする．言語

$$B = \{ w \mid w \text{ の下段は上段の 3 倍} \}$$

とする．ただし空列 ε も B に含まれるとする．たとえば，

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in B$$

である．このとき，B の各要素を逆に読んだ文字列からなる言語

$$B^{\mathcal{R}} = \{ w^{\mathcal{R}} \mid w \in B, \text{ただし}, w^{\mathcal{R}} \text{ は } w \text{ を逆から読んだ文字列} \}$$

を認識する 4 状態の決定性有限オートマトンの状態遷移図を与えよ．

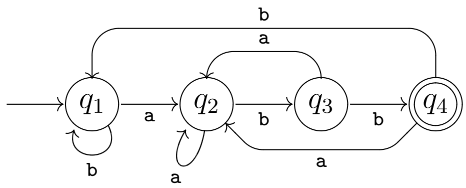
（次ページに続く）

注 1:　**ポンピング補題**　言語 L が正規言語であるとき，以下のような数 p (ポンピング長) が存在する:

> s が $|s| \geq p$ であるような L の任意の文字列であるとき，s は次の条件を満たすように 3 つの部分 $s = xyz$ に分割できる:
>
> (a) 各々の $i \geq 0$ に対して $xy^iz \in L$
>
> (b) $|y| > 0$
>
> (c) $|xy| \leq p$

ただし，$|s|$ は文字列 s の長さを表わし，y^i は y を i 個連結したものを表わす．y^0 は空列 ε (文字を 1 つも含まない文字列) となる．

注 2:　**決定性有限オートマトンの状態遷移図**　以下は，アルファベットが $\{\mathtt{a}, \mathtt{b}\}$ であるような言語 $\{w \mid w$ は abb で終わる $\}$ を認識する決定性有限オートマトンの状態遷移図の例である．開始状態は q_1，受理状態は q_4 である．



Problem 7

Let

$$\Sigma = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}.$$

Answer the following questions on languages over Σ.

(1) Let

$$A = \{w \mid \text{the upper row string of } w \text{ is a palindrome}\},$$

where a palindrome is a string that reads the same forward and backward. The language A includes the empty string ε. For example,

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in A.$$

Show that the language A is not regular by using the pumping lemma.

(2) Give a context-free grammar that generates the language A.

(3) Here, consider each row of a string over alphabet Σ to be a binary number, with the left most position indicating the most significant bit. Let

$$B = \{w \mid \text{the lower row of } w \text{ is three times the upper row}\}.$$

The language B includes the empty string ε. For example,

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in B.$$

Let

$$B^{\mathcal{R}} = \{w^{\mathcal{R}} \mid w \in B, \text{where } w^{\mathcal{R}} \text{ is a string that reads backward of } w\}.$$

Give a state diagram of a deterministic finite automaton for recognizing the language $B^{\mathcal{R}}$, by using four states.

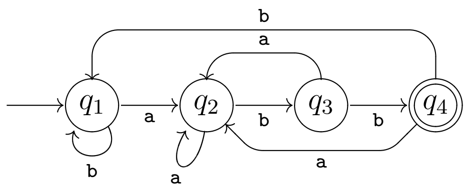
(Continued on the next page)

Note 1: **The pumping lemma** If L is a regular language, then there is a number p (the pumping length) such that

> if s is any string in L such that $|s| \geq p$, then s can be divided into three pieces, $s = xyz$, satisfying the following conditions:
>
> (a) for each $i \geq 0$, $xy^iz \in L$,
>
> (b) $|y| > 0$, and
>
> (c) $|xy| \leq p$,

where $|s|$ represents the length of the string s, y^i means that i copies of y are concatenated together, and y^0 equals the empty string ε (the string of the length zero).

Note 2: **A state diagram of a deterministic finite automaton** Consider the alphabet $\{\mathtt{a}, \mathtt{b}\}$. A state diagram of a finite automaton recognizing the language $\{w \mid w \text{ ends with } \mathtt{abb}\}$ is as follows, where q_1 is the initial state, and q_4 is the accepting state.



問 8

図 1 の二分木のように最大の深さを除き節点が完全に詰まっており，また，深さ最大の葉が左詰めになっている二分木を広義の完全二分木と言う．広義の完全二分木の各節点に図 1 に示したように番号を付け，図 2 のように配列で表現する．たとえば，10 が格納されている節点には番号 2 が付いており, その配列表現において $A[2] = 10$ となっている．配列 A が次の条件を満たす時，この配列 A 及び対応する広義の完全二分木をヒープと呼ぶ．

（ヒープ条件）根以外の任意の節点 i について，$A[\mathrm{parent}(i)] \geq A[i]$ が成り立つ．

ここで，$\mathrm{parent}(i)$ は節点 i の親の番号とする．図 2 の配列 A はヒープ条件を満たしている．ヒープの各節点の高さを，その節点から葉までの最長の経路の辺の数とする．たとえば，図 1 において，2 が格納されている節点 9 は葉なので高さは 0 であり，節点 1 の高さは 2 であり，根（節点 0）の高さは 3 である．節点の高さについて，以下の二つの性質が成り立つ．

(P1) n 個の要素を持つヒープの根の高さは，$\lfloor \log_2 n \rfloor$ である．

(P2) n 個の要素を持つヒープにおいて，高さ h の節点は $\lceil n/2^{h+1} \rceil$ 個以下である．

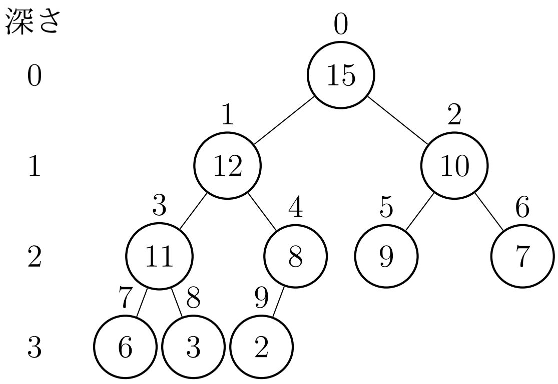


図 1: ヒープ（二分木表現）

配列 A

0	1	2	3	4	5	6	7	8	9
15	12	10	11	8	9	7	6	3	2

図 2: ヒープ（配列表現）

（次ページに続く）

次に，配列からヒープを構成する下のアルゴリズムを考える．関数 build_heap(a,n) は大きさ n の配列 a をヒープに変換する．補助関数 heapify(a,i,n) は節点 i の左右の部分木がヒープ条件を満たしているときに適用され，節点 i を根とする部分木をヒープにする．

```
void heapify(int a[], int i, int n) {
  int k = 2*i+1;                      /* 節点 k は，節点 i の左の子 */
  while (k < n) {
    if (k+1 < n && a[k+1] > a[k]) k = k+1;
    if (a[i] >= a[k]) return;
    int tmp = a[i];
    a[i] = a[k];
    a[k] = tmp;
    i = k; k = 2*i+1;
  }
}

void build_heap(int a[], int n) {
  for (int i = n-1; i >= 0; i--) {
    heapify(a, i, n);
  }
}
```

以下の問に答えよ．

(1) 1 から 7 までの整数を 1 つずつ含む要素数 7 のヒープ（配列表現）が何種類あるか示せ.

(2) 配列 $a_1 = \{2, 6, 7, 3, 9, 11, 8, 10, 15, 4\}$ に対して，build_heap(a_1,10) を実行する．実行後の配列 a_1 の要素を順番に書け.

(3) 要素数 n の配列 a とその節点 i に対して，heapify(a,i,n) を実行する．節点 i の高さを h とすると，実行時間は $O(h)$ となる．その理由を説明せよ．

(4) 大きさ n の配列 a に対する build_heap(a,n) の実行時間が $O(n)$ であることを，性質 (P1), (P2) を用いて示せ． $\displaystyle\sum_{k=1}^{m} \frac{k}{2^k} = 2 - \frac{m+2}{2^m}$ を使用して良い．

Problem 8

A nearly complete binary tree is a binary tree that is filled on all levels except possibly the deepest, which is filled from the left, as shown in Figure 1. Each node of a nearly complete binary tree is indexed as shown in Figure 1 and then the tree can be represented by the array A in Figure 2. For example, the node with value 10 has index 2 and thus $A[2] = 10$ in the array representation. The array A and also the binary tree corresponding to it are called heaps if the following heap property is satisfied:

（Heap Property） for every node i other than the root, $A[\text{parent}(i)] \geq A[i]$ holds

where parent(i) is the index of the parent of node i. The array A in Figure 2 satisfies the heap property.
We define the height of a node to be the number of edges on the longest path from the node to a leaf. For the heap in Figure 1, node 9 that stores 2 is a leaf and thus its height is 0. The height of node 1 is 2, and the height of the root (node 0) is 3. Heaps satisfy the following two properties on the height of a node.

(P1) The root of an n-element heap has height $\lfloor \log_2 n \rfloor$.

(P2) The number of nodes of height h in an n-element heap is bounded by $\lceil n/2^{h+1} \rceil$.

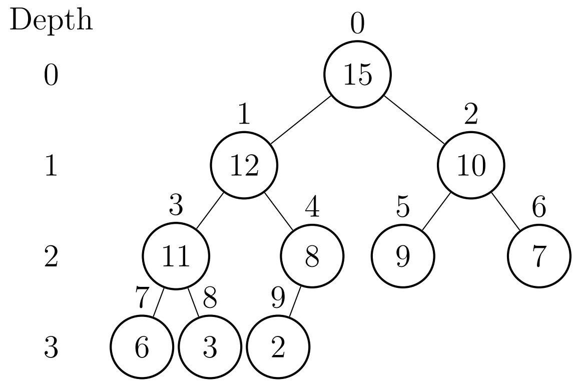


Figure 1: Heap (binary tree representation)

Array A

0	1	2	3	4	5	6	7	8	9
15	12	10	11	8	9	7	6	3	2

Figure 2: Heap (array representation)

(Continued on the next page)

Next, we consider the following algorithm that constructs a heap from an arbitrary array. The function `build_heap(`a`,`n`)` converts an n-element array a into a heap. The auxiliary function `heapify(`a`,`i`,`n`)` is applied to a node i when both of the left and right subtrees of i satisfy the heap property and converts the subtree rooted at the node i into a heap.

```
void heapify(int a[], int i, int n) {
  int k = 2*i+1;               /* node k is the left child of node i */
  while (k < n) {
    if (k+1 < n && a[k+1] > a[k]) k = k+1;
    if (a[i] >= a[k]) return;
    int tmp = a[i];
    a[i] = a[k];
    a[k] = tmp;
    i = k; k = 2*i+1;
  }
}

void build_heap(int a[], int n) {
  for (int i = n-1; i >= 0; i--) {
    heapify(a, i, n);
  }
}
```

Answer the following questions.

(1) Show the number of different 7-element heaps (array representation) that contain all integers from 1 to 7.

(2) For $a_1 = \{2, 6, 7, 3, 9, 11, 8, 10, 15, 4\}$, we run `build_heap(`$a_1$`,10)`. Write down the elements of the array a_1 after the execution.

(3) For an n-element array a and its node i, we run `heapify(`a`,`i`,`n`).` Then, its running time is $O(h)$ for a node i of height h. Explain the reason.

(4) Show that the running time of `build_heap(`a`,`n`)` is $O(n)$ by using the properties (P1) and (P2) where n is the size of the array a. You can use $\displaystyle\sum_{k=1}^{m} \frac{k}{2^k} = 2 - \frac{m+2}{2^m}$.

問 9

クリティカルセクション（critical section，以下 CS）とはプログラム領域であり，同時には 1 個のスレッドだけがその実行を許される．これを達成するように CS を囲む前処理と後処理を設計することを，クリティカルセクション問題と呼ぶ．
プログラム 1 とプログラム 2 のように，クリティカルセクション問題の解決を試みた．以下，2 個のスレッド，スレッド 0 とスレッド 1 だけが存在し，両スレッドともに同じプログラム（プログラム 1 または 2）を実行するとする．スレッドは前処理以下を複数回実行する可能性があることに注意せよ．プログラム 1，2 において，2 個のスレッドは変数 `turn` と `flag` を共有しており，一方のスレッドによる変更は他方のスレッドでも即座に観測されるとする．スレッド作成前に，`turn` には 0 が，`flag[0]` と `flag[1]` には `false` が代入される．変数 `i` と `j` は次の値をとる．スレッド 0 では `i = 0` および `j = 1`，スレッド 1 では `i = 1` および `j = 0`．

<u>プログラム 1</u>:

```
...
while (turn != i) ;  // 待つ          ··· 前処理
＜クリティカルセクションのコード＞
turn = j;                            ··· 後処理
...
```

<u>プログラム 2</u>:

```
...
flag[i] = true; // このスレッドが CS に入ろうとしている
turn = j;       // 別のスレッドを優先する                  ··· 前処理
while (flag[(a)] && turn == (b)) ;  // 待つ
＜クリティカルセクションのコード＞
flag[(c)] = false;                                        ··· 後処理
...
```

(1) この状況で，クリティカルセクション問題の解決が満たすべき性質は次の 3 つである．

A. 相互排除 : CS に入っているスレッドがある場合，他のスレッドは CS に入れない．

B. 進行 : CS に入っているスレッドがなく，1 個以上のスレッドが前処理を実行する場合，それらのうち 1 個は CS に入れる．

C. 有限の待ち回数 : 前処理の実行を始めてから CS に入るまでの間に，他のスレッドの方が先に CS へ入ることもある．しかしその回数には上限がある．

しかし，プログラム 1 は性質 A〜C のうち 1 つを満たしていない．プログラム 1 が満たしていない性質はどれか，答えよ．また，その理由を答えよ．

(2) プログラム 2 は，性質 A〜C のすべてを満たす．(a)，(b)，(c) にはそれぞれ `i` か `j` が入る．どちらが入るか，答えよ．

（次ページに続く）

(3) プログラム 1 や 2 のようにスレッド自身が変数の値を調べ続ける方法はビジーウェイト（busy wait）やポーリング（polling）と呼ばれる．一方で，スレッドを実行待ち状態（waiting state）にする方法もある．

消費電力について，2 つの方法のどちらが優れているか，答えよ．また，その理由を数行程度で述べよ．

Problem 9

A critical section (CS) is a segment of a program that at most one thread is allowed to execute at the same time. The critical-section problem is to design an entry section and an exit section that surround a CS to achieve it.
We try to solve the critical-section problem as shown in Program 1 and Program 2. Assume that there are two threads, namely thread 0 and thread 1, that execute the same program (Program 1 or 2). Note that a thread may execute an entry section and following lines multiple times. In Program 1 and Program 2, the two threads share variables `turn` and `flag`. Their updates by a thread are immediately observed by the other thread. Before the two threads are created, `turn` is set to `0`, and `flag[0]` and `flag[1]` are set to `false`. Variables `i` and `j` hold the following values. In thread 0, `i` = `0` and `j` = `1`. In thread 1, `i` = `1` and `j` = `0`.

Program 1:

```
...
while (turn != i) ;  // wait     ··· Entry section
<critical section>
turn = j;                        ··· Exit section
...
```

Program 2:

```
...
flag[i] = true; // this thread tries to enter a CS
turn = j;    // gives way to another thread                    ··· Entry section
while (flag[(a)] && turn == (b)) ;  // wait
<critical section>
flag[(c)] = false;               ··· Exit section
...
```

(1) In this situation, a solution to the critical-section problem should satisfy the following three properties.

A. Mutual exclusion: If a thread is in the CS, no other threads can be in the CS.

B. Progress: If no thread is in the CS and one or more threads execute the entry section, one of the threads enters the CS.

C. Bounded waiting: There exists a bound on the number of times that other threads enter the CS after a thread starts executing the entry section and before it enters the CS.

Program 1 does not satisfy one property out of the three properties. Answer which property is not satisfied. Describe its reason.

(2) Program 2 satisfies all the three properties. Fill (a), (b) and (c) with `i` or `j`.

(Continued on the next page)

(3) The approaches that Program 1 and 2 take is called “busy wait” or “polling,” in which a thread keeps running and repeatedly checks the value of a variable. On the other hand, there is another approach in which a thread transitions to the “waiting state.”

Which of the two approaches is superior in terms of power consumption? Explain its reason in a few lines.