専門科目 (午前)

数理・計算科学

15　大修

時間 午前 9 時 30 分 – 午後 1 時

注意

1. 専門基礎問題 [1], [2], [3] より 2 問を選択 し解答せよ.
2. 専門一般問題 [4],. . ., [12] より 3 問を選択 し解答せよ.
3. 解答は 1 題ごとに 別々の解答用紙に記入 せよ.
4. 要求されたより多くの問題に解答した場合は 採点されない 可能性がある.
5. 解答用紙毎に必ず 問題の番号および受験番号 を記入せよ.

問 1 (基礎問題)

行列 $A = \begin{pmatrix} 4 & 2 \\ 1 & 5 \end{pmatrix}$ を考える．

(a) 行列 A の固有値と対応する右固有ベクトルを求めよ．

(b) 行列 e^A を $I + \sum_{n=1}^{\infty} \frac{1}{n!} A^n$ で定義する．ここで I は単位行列である．この行列 e^A の固有値と対応する右固有ベクトルを求めよ．

(c) $n \geq 1$ に対して，A^n の各要素を n の関数として表せ．

(d) 行列 e^A の各要素を具体的に書き表せ．

問 2 (基礎問題)

正の整数 n と実数 $t > 0$ に対して，関数列 $(I_n(t))$ を

$$I_n(t) = \int_0^\infty e^{-tx} \sin(nx)\, dx$$

によって定める.

(1) $I_n(t)$ を求めよ.

(2) $\lim_{n\to\infty} I_n(t)$ を求めよ．この収束は t について一様かどうかを答えよ.

問 3 (基礎問題)

次のように定義されるオーダー記法について，以下の問いに答えよ．なお，以下に出てくる関数は，すべて，自然数から非負実数への関数とする．

> **定義.** 関数 f, t に対し，次の条件が成り立つとき,「$f(n) = O(t(n))$ である」といい，成り立たないとき,「$f(n) \neq O(t(n))$ である」という．
>
> $$\exists c_0,\ \exists n_0,\ \forall n \geq n_0\ [\ f(n)\ \leq\ c_0 \cdot t(n)\ ].$$

(1) $f(n) = O(n^2)$, $g(n) = O(1)$ となる関数に対し，関数 h を $h(n) = f(n) \cdot g(n)$ と定義する．このとき，$h(n) = O(n^2)$ となることを示せ．

(2) $f(n) \neq O(n^2)$ かつ，$g(n) \neq O(1)$ となる関数でも，$h(n) = f(n) \cdot g(n)$ と定義すると，$h(n) = O(n^2)$ となる場合がある．そのような関数 f, g の例を示せ．

問 4 (一般問題)

位数 n の巡回群を $\mathbb{Z}/n\mathbb{Z}$ で表し，$G = \mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/6\mathbb{Z}$ とする．このとき次の設問に答えよ．

(1) G の位数 2 の部分群を列挙せよ（三つある）．

(2) G の指数 2 の部分群を列挙せよ（三つある）．

(3) G は $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/12\mathbb{Z}$ に同型であることを示せ．

(4) G の位数 2 の部分群を H_1，H_2，H_3 とするとき，G/H_1，G/H_2，G/H_3 は互いに同型であるかどうかを調べよ．

問 5 (一般問題)

H を実ヒルベルト空間とする．このとき，つぎの (1)，(2) を証明せよ．

(1) x，$y \in H$ と実数 α に対して

$$\|\alpha x + (1-\alpha)y\|^2 = \alpha\|x\|^2 + (1-\alpha)\|y\|^2 - \alpha(1-\alpha)\|x-y\|^2$$

が成り立つ．

(2) C を H の閉凸集合とし，T を C から C への非拡大写像とする．すなわち，任意の x，$y \in C$ に対して

$$\|Tx - Ty\| \leq \|x - y\|$$

を満たすものとする．このとき，T の不動点からなる集合 $F(T)$ は閉凸集合である．

問 6 (一般問題)

次の設問に答えよ.

(1) $[0,\infty)\times\mathbb{R}$ 上で 2 回連続微分可能な関数 $U(t,x)$ が

$$U_{tt} = U_{xx} \quad (\ t>0,\ x\in\mathbb{R}\ )$$
$$U(0,x) = F(x), \quad U_t(0,x) = G(x) \quad (\ x\in\mathbb{R}\ )$$

を満たすとする. $U(t,x)$ を $F(x)$ と $G(x)$ を用いて表せ.

(2) $[0,\infty)\times\mathbb{R}$ 上で 2 回連続微分可能な関数 $u(t,x)$ が

$$u_{tt} + 2u_t + u = u_{xx}, \quad (t>0,\ x\in\mathbb{R})$$
$$u(0,x) = f(x), \quad u_t(0,x) = g(x) \quad (\ x\in\mathbb{R}\ )$$

を満たすとする. $u(t,x)$ を $f(x)$ と $g(x)$ を用いて表せ. さらに, $f(x)$ と $g(x)$ がともに有界であるならば,

$$u(t,x)\to 0 \quad (\ t\to\infty\ )$$

であることを示せ. ($u(t,x) = e^{-t}U(t,x)$ とおけ.)

問 7 (一般問題)

$\boldsymbol{a}_i \in \mathbb{R}^n$ を定数列ベクトル, $b_i \in \mathbb{R}$ を定数とする $(i = 1, 2, \ldots, m)$．P を

$$P = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}_i^T \boldsymbol{x} \le b_i \ (i = 1, 2, \ldots, m)\}$$

で定義し，

(a) P は空でない

(b) P は有界

を仮定する．ただし，$\mathbb{R}^n$ は n 次元 Euclid 空間，T はベクトルの転置を表す．このとき，以下の問 (i), (ii), (iii) に答えよ.（ヒント：仮定 (a) と (b) のもとでは，P は n 次元 Euclid 空間内の多面体になる．$n = 2$ の場合について図を描いて考えるとよい.）

(i) 「$\boldsymbol{a}_i^T \boldsymbol{d} \le 0 \ (i = 1, 2, \ldots, m)$ を満たす非ゼロベクトル $\boldsymbol{d} \in \mathbb{R}^n$ が存在しない」ことを証明せよ.

(ii) $\boldsymbol{v} \in P$ が，条件

$$\boldsymbol{x}^1 \in P,\ \boldsymbol{x}^2 \in P,\ \boldsymbol{v} = \frac{1}{2}\boldsymbol{x}^1 + \frac{1}{2}\boldsymbol{x}^2 \implies \boldsymbol{x}^1 = \boldsymbol{x}^2$$

を満たすとき，$\boldsymbol{v}$ は P の頂点であるという．また，任意の $\boldsymbol{x} \in P$ に対して，$\{1, 2, \ldots, m\}$ の部分集合 $I(\boldsymbol{x})$ を $I(\boldsymbol{x}) = \{i : \boldsymbol{a}_i^T \boldsymbol{x} = b_i\}$ で定義する.「$\boldsymbol{x} \in P$ が P の頂点でなければ，ある非ゼロベクトル $\boldsymbol{d} \in \mathbb{R}^n$ と $k \notin I(\boldsymbol{x})$ が存在して，

$$\boldsymbol{a}_i^T \boldsymbol{d} = 0 \ (i \in I(\boldsymbol{x})),\ \boldsymbol{a}_k^T \boldsymbol{d} > 0$$

を満たす」ことを証明せよ.

(iii) 任意の $\boldsymbol{x} \in P$ に対して，$I(\boldsymbol{x})$ は高々 n 個の要素からなることを仮定する．このとき,「$\boldsymbol{x} \in P$ において，$I(\boldsymbol{x})$ がちょうど n 個の要素からなれば $\boldsymbol{x}$ は P の頂点である」ことを証明せよ.

問 8 (一般問題)

次の問に答えよ．

(1) 確率変数の系列 $U_1, U_2, \ldots$ を考え，これらが互いに独立に $(0,1)$ 上の一様分布に従うとする．確率変数 $X_n = \max\{U_1, U_2, \ldots, U_n\}$ の分布関数 $F_n(x) = \Pr\{X_n \leq x\}$ を求めよ．

(2) (1) の X_n を用い $Y_n = n(1 - X_n)$ とおく．Y_n の分布関数 $G_n(y) = \Pr\{Y_n \leq y\}$ を求め，さらに Y_n の極限分布関数 $G(y) = \lim_{n\to\infty} G_n(y)$ を求めよ．

(3) 確率変数の系列 $V_1, V_2, \ldots$ を考え，これらが互いに独立に確率密度関数 $f_V(v) = \exp(-v)$, $v > 0$ の指数分布に従うとき，

$$Z_n = -\max\{V_1, V_2, \ldots, V_n\} + \log n$$

の極限分布関数 $H(z) = \lim_{n\to\infty} \Pr\{Z_n \leq z\}$ を求めよ．

問 9 (一般問題)

確率変数 $X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_n$ は同じ平均値 μ と分散 σ^2 を持つ．これらの変数の共分散に付いては次の関係が成り立つものとする

$$\mathrm{Cov}\{X_i, X_j\} = \mathrm{Cov}\{Y_i, Y_j\} = 0 \quad (i \neq j \text{ の時})$$

$$\mathrm{Cov}\{X_i, Y_j\} = \begin{cases} 0 & (i \neq j \text{ の時}) \\ -\sigma^2/2 & (i = j \text{ の時}) \end{cases}$$

(1) X, Y データの標本平均をそれぞれ $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$，$\bar{Y} = \frac{1}{n}\sum_{j=1}^{n} Y_j$ とするとき，分散 $\mathrm{Var}\{\bar{X}\}$, $\mathrm{Var}\{\bar{Y}\}$ と共分散 $\mathrm{Cov}\{\bar{X}, \bar{Y}\}$ を求めよ．

(2) μ の不偏推定量 $Z = c\bar{X} + (1-c)\bar{Y}$ の分散が最小になる定数 c とそのときの分散の値を求めよ．

問 10 (一般問題)

以下の問のプログラム中では，i, j, k を整数型の変数，a を十分に大きな整数型の配列とする．そして，任意の非負整数 m に対し，a[m] の参照および a[m] への代入には，それぞれ $m+1$ 単位時間を要するものとする．一方，その他の計算（四則演算，大小比較，制御命令，i, j, k の読み書きなど）は非常に高速で，要する時間を無視してよいと仮定する．

(1) 次のプログラム断片は a[0], . . . , a[$n-1$] に格納された n 個の要素 ($n \geq 1$) をソートするものである．このプログラム断片の実行に要する時間が最大となるのは，a[0], . . . , a[$n-1$] がどのような条件を満たす時か述べよ．また，その時の実行に要する時間を求めよ．結果は，オーダー記法ではなく，n の多項式の形で書き表すこと．

```
for(i = 0; i < n - 1; i++)
    for(j = i + 1; j < n; j++)
        if(a[j] < a[i]) {
            k = a[i];
            a[i] = a[j];
            a[j] = k;
        }
```

(2) n 個 (ただし，$n = 2^m - 1$，m は正整数) のソートされたデータが a[0], . . . , a[$n-1$] に格納されているものとする．これらのデータのうちの一つをキー (key) として二分探索を行った場合の平均の探索時間を求めよ．平均探索時間を求めるにあたっては，a[0], . . . , a[$n-1$] までの各要素がすべて等確率で探索され，他の値が探索されることはないと仮定すること．なお，二分探索アルゴリズムの挙動については，以下のプログラム断片を参考にせよ．

```
i = 0;
j = n - 1;
while(true) {
    k = a[(i + j)/2];
    if(key == k)
        break;
    else if(key < k)
        j = (i + j)/2 - 1;
    else
        i = (i + j)/2 + 1;
}
```

問 11 (一般問題)

プログラミング言語Cで書かれたソースプログラム

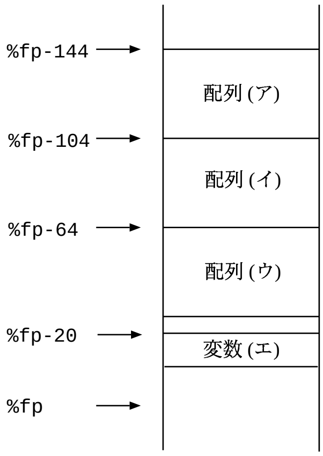
```
int i ;
int a[10], b[10], c[10] ;   /* a, b, c は添字が 0..9 の整数の配列 */
for ( i = 0 ; i < 10 ; i = i + 1 ) /* i を 0 から 9 まで繰り返す */
   a[i] = b[i] - c[i] ;
```

をコンパイルしたところ，以下の目的コード（アセンブリ言語形式）が得られた.（ソースプログラム，目的コードとも該当部分のみ示す）．最適化は行っていない．以下の目的コードの行頭の数字は行番号であって目的コードの一部ではない.

```
43  .LL3:
44          st      %zero, [%fp-20]
45  .LL6:
46          ld      [%fp-20], %r1
47          cmp     %r1, 9
48          ble     .LL9
49          nop
50          b       .LL7
51          nop
52  .LL9:
53          ld      [%fp-20], %r1
54          sll     %r1, 2, %r6
55          add     %fp, -64, %r5
56          ld      [%fp-20], %r1
57          sll     %r1, 2, %r4
58          add     %fp, -104, %r2
59          ld      [%fp-20], %r1
60          sll     %r1, 2, %r3
61          add     %fp, -144, %r1
62          ld      [%r2+%r4], %r2
63          ld      [%r1+%r3], %r1
64          sub     %r2, %r1, %r1
65          st      %r1, [%r5+%r6]
66          ld      [%fp-20], %r1
67          add     %r1, 1, %r1
68          st      %r1, [%fp-20]
69          b       .LL6
70          nop
71  .LL7:
```

43 行目～52 行目と 66 行目～70 行目は「`for ( i = 0 ; i < 10 ; i = i + 1 )`」に対する目的コード，53 行目～65 行目は,「`a[i] = b[i] - c[i] ;`」に対する目的コードである．付録につけた説明を参考に，まずこの目的コードの 53 行目～65 行目を読んで，次の問に答えよ.

(1) この目的コードでは，実行時に配列 a,b,c や変数 i がメモリの活性レコード（フレーム）上に取られることになっている（下図）．番地はバイト単位につけられている.



レジスタ`%fp`はフレームポインタと呼ばれるものであるが，どのような役割をしているか，簡単に述べよ．

また，この図の（ア）〜（エ）が何かを答えよ．

(2) 行 56, 57, 58, 62 は`b[i]`の値を取り出すコードである．

配列の各要素が何バイトを占めるか答えよ．また，この４行のコードについて，行ごとに何を計算しているか簡単に説明せよ．とくに，57 行目で乗算命令でなく`sll` (shift left logical) 命令を使う理由を答えよ．

(3) この目的コードにはいろいろな無駄がある．たとえば，53 行目〜65 行目は「`a[i] = b[i] - c[i] ;`」の目的コードであるが，同じような計算を何度も行っている．レジスタの使い方ももっと工夫ができそうである．

この目的コードの 53 行目〜65 行目に対し，どのような最適化ができそうか，この目的コードに沿って述べよ．

さらにこの目的コードの全体を読み，どのような最適化ができるか述べよ．

（付録）このアセンブリ言語の読み方

`%`はレジスタを表す．レジスタにはいくつもの種類があるが，さしあたり汎用レジスタ`%r1`から`%r7`と，特殊な使い方をする`%fp`, およびつねに値 0 をもつ`%zero`があると思えばよい．命令形式の [...] は「... 番地の内容」を表す．

命令形式	意　味
`ld [%reg1 + %reg2], %reg3` または `ld [%reg1 +- const], %reg3`	(`%reg1 + %reg2`) 番地または (`%reg1 +- const`) 番地の内容を`%reg3` にロードする．
`st %reg1, [%reg2 + %reg3]` または `st %reg1, [%reg2 +- const]`	`%reg1` を (`%reg2 + %reg3`) 番地または (`%reg2 +- const`) 番地へストアする．
`add` または `sub` `%reg1, %reg2` または `const, %reg3`	`%reg1 +`または`-` (`%reg2` または `const`) の結果を`%reg3` へしまう．
`sll %reg1, const, %reg2`	`%reg1` を `const` ビット左シフトしたものを`%reg2` へしまう．
`cmp %reg1, const`	`%reg1` と `const` を比較する．condition code が立つ．
`ble .L` または `bl .L`	比較の結果が≦または＜なら`.L` 番地へ飛ぶ．
`b .L`	無条件に`.L` 番地へ飛ぶ．
`nop`	遅延分岐用の無演算命令だが忘れてよい．

問 12 (一般問題)

下に示す getNext は，整数カウンターを実現するプログラムである．このプログラムは 2 つの変数 a, b (C 言語なら大域変数，Java 言語ならフィールドと考えよ) を使い，getNext が実行された回数をそれらの変数に保存する．変数 a には getNext が引数 0 で実行された回数が保存され，変数 b には getNext が 0 以外の引数で実行された回数が保存される．複数のスレッドが同じ引数で同時に getNext を呼んでも，保存される実行回数が正しくなるよう，排他制御が必要である．そこで，下のプログラムでは critical section を表わす CS { ... } という構文を使っている．{ ... } の部分は，一度に 1 つのスレッドしか実行することができないとする．CS 文がプログラム中に複数個ある場合は，それぞれについて 1 つのスレッドが実行できる．

```
int a = 0;
int b = 0;

int getNext(int s) {
    int result;
    CS {
        if (s == 0) {
            a = a + 1;
            result = a;
        }
        else {
            b = b + 1;
            result = b;
        }
    }

    return result;
}
```

(1) 上のプログラムで CS 構文を使わないと，複数のスレッドが同じ引数で同時に getNext を呼び出したときに競合状態 (race condition) が発生する可能性がある．競合状態とは何か，上のプログラムを例に，具体的に説明せよ．

(2) CS 文の機能が言語処理系によって正しく実装されていないと，多数のスレッドが getNext を呼び出したとき，一部のスレッドに飢餓状態 (starvation) が発生する可能性がある．飢餓状態とは何か説明せよ．

(3) 上のプログラムでは，片方のスレッドが getNext(0) を，他方のスレッドが getNext(1) を同時に呼び出した場合も，どちらか一方のスレッドしか CS 文を即座に実行できず，他方は待たされる．しかし，それらの呼び出しは互いに影響を与えないので，本来，同時並行的に実行できるはずである．

プログラムを書き変えて，そのような場合には同時並行的に 2 つのスレッドが getNext を実行できるようにせよ．