平 成 １９年 度

名古屋大学大学院情報科学研究科
情報システム学専攻
入 学 試 験 問 題

専　　　　門

平成１８年８月８日（火）
１２：３０～１５：３０

注 意 事 項

1．試験開始の合図があるまでは、この問題冊子を開いてはならない。
2．試験終了まで退出できない。
3．外国人留学生は、和英辞書などの辞書を１冊に限り使用してよい。
電子辞書の持ち込みは認めない。
4．問題冊子、解答用紙４枚、草稿用紙４枚が配布されていることを確認せよ。
5．問題は、A、Bの各科目群について下記のように解答せよ（合計４科目を選択して解答せよ）。
A群：次の３科目から２科目を選択して解答せよ。
(1) 解析・線形代数　(2) 確率・統計　(3) プログラミング
B群：次の３科目から２科目を選択して解答せよ。
(1) 計算機理論　(2) ハードウェア　(3) ソフトウェア
なお、選択した科目名を解答用紙の指定欄に記入せよ。
B群において計算機理論を選択する場合は、計算機理論Iと計算機理論IIのいずれか一方を選択して解答せよ。計算機理論Iと計算機理論IIの両方を選択することはできない。解答用紙の指定欄には計算機理論Iまたは計算機理論IIと記入せよ。
6．解答用紙は、指定欄に受験番号を必ず記入せよ。解答用紙に受験者の氏名を記入してはならない。
7．解答用紙は、試験終了後に４枚とも提出せよ。
8．問題冊子、草稿用紙は、試験終了後に持ち帰ってよい。

解析・線形代数

（解の導出過程も書くこと）

[1]　$C: x^2 + y^2 = 1$を正の向きにとる時，線積分

$$\int_C (x^2 y)dx - (xy^2)dy$$

を求めよ．

[2]　2次形式$f = x_1^2 + 3x_2^2 + 2x_3^2 + 4x_2x_3 + 4x_3x_1$を考える．以下の問いに答えよ．

(1)　$f = \mathbf{x}^T \mathbf{A} \mathbf{x} = \sum_{i,j=1}^{3} a_{ij} x_i x_j$と表すとき，行列$\mathbf{A}$を求めよ．ただし，$\mathbf{x} = (x_1, x_2, x_3)^T$であり，$a_{ij}$は行列$\mathbf{A}$の$i$行$j$列の要素である．

(2) 行列$\mathbf{A}$の固有値$\lambda_1, \lambda_2, \lambda_3$（ただし，$\lambda_1 > \lambda_2 > \lambda_3$）と対応する単位固有ベクトルを求めよ．

(3) 行列$\mathbf{A}$を$\mathbf{P}^T\mathbf{A}\mathbf{P}$で対角化するとき，行列$\mathbf{P}$ならびに対角化された行列$\mathbf{Q}$を求めよ．

(4) 変数変換$\mathbf{x} = \mathbf{P}\mathbf{y}$（$\mathbf{y} = (y_1, y_2, y_3)^T$）により，$f$を$y_1, y_2, y_3$を用いて表せ．

(5)　$x_1^2 + x_2^2 + x_3^2 = 1$のとき，$f = x_1^2 + 3x_2^2 + 2x_3^2 + 4x_2x_3 + 4x_3x_1$の最大値，最小値を求めよ．

注:

線積分	line integral	行列	matrix	転置	transpose
対角化	diagonalization	固有値	eigenvalue	単位固有ベクトル	unit eigenvector

確率・統計（解の導出過程を書くこと）

[1] 確率変数 Xの確率分布関数を$F(x)$とし，確率変数Yを$Y = X^2$とする．下記の問いに答えよ．

(1) 確率変数Yの確率分布関数 $G(y)$を，Fを用いて表わせ．

(2) 確率変数Xの確率密度関数$f(x)$が次式であるとき，確率変数Yの確率密度関数$g(y)$を求めよ．

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{x^2}{2\sigma^2})$$

[2] 互いに独立な確率変数X, Yがあり，確率変数Zは$Z = X + Y$とする．以下の問いに答えよ．

(1) X, Yがともに，区間$[-\frac{1}{2}, \frac{1}{2})$における一様分布に従うとき，$Z$の確率密度関数$h(z)$を求めよ．

(2) X, Yがともに正規分布$N(\mu, \sigma^2)$に従うとする．次の問いに答えよ．

(a) Xのモーメント母関数（積率母関数）を求めよ．

(b) Zのモーメント母関数を求め，Zも正規分布となることを示せ．

[専門用語の英訳]

確率変数 random variable

確率分布関数 probability distribution function

確率密度関数 probability density function

独立 independence

一様分布 uniform distribution

モーメント母関数（積率母関数） moment generation function

正規分布 normal distribution

プログラミング

[1] 以下に、C 言語による、先入れ先出し (First-In First-Out) 待ち行列 (queue) を実現するプログラムを示す。このプログラムに関する以下の問に答えよ。なお、プログラムの左側の数字は、行の番号を示すもので、プログラムの一部ではない。

```
 1:    #include <stdio.h>
 2:
 3:    #define SIZE    10
 4:    #define INC(x)  [          (a)          ]
 5:
 6:    int queue[SIZE];
 7:    int head, tail, count;
 8:
 9:    void emptyQueue() {
10:        head=tail=count=0;
11:    }
12:
13:    void enQueue(int x) {
14:        if (count < SIZE) {
15:            count++;
16:            queue[tail] = x;
17:            INC(tail);
18:        }
19:    }
20:
21:    int deQueue() {
22:        [          (b)          ]
23:    }
24:
25:    int main() {
26:        emptyQueue();
27:
28:        enQueue(10);
29:        enQueue(30);
30:        printf("%d\n", deQueue());
31:        enQueue(50);
32:        enQueue(20);
33:        enQueue(60);
34:        printf("%d\n", deQueue());
35:        enQueue(40);
36:
37:        return 0;
38:    }
```

（1） $\boxed{\text{(a)}}$ には、待ち行列を操作したときに変数 `head` または `tail` を更新するためのマクロ `INC(x)` の定義が入る。変数 `head`, `tail` はそれぞれ配列 `queue` における待ち行列の先頭位置、最後尾位置を管理するための値を保持する。配列 `queue` の要素は、順に待ち行列のデータの記憶に使用されるが、最後の要素が使用された後は再び先頭から空き要素が順に使われるようにする。このような配列の使用法を環状リスト (cyclic list) (あるいは巡回バッファ (ring buffer)) という。本プログラムが、環状リストの実現になるように、`INC(x)` の定義を書け。

（2） $\boxed{\text{(b)}}$ に記述されるべき関数 `deQueue()` は、待ち行列から先頭のデータを取り出す関数である。以下の要件を満たす関数定義を書け。記述は複数行になってよい。

- 待ち行列の先頭のデータを関数値として返す。
- 待ち行列の先頭のデータを待ち行列から除く。ただし、配列 `queue` の要素には変更を加えない。
- 待ち行列が空 (empty) の場合は値 `-1` を返す。
- `head`, `tail`, `count` を適切に更新する。

（3） `main()` 関数の 35 行目実行後の時点での、`queue`, `head`, `tail`, `count` の値を示せ。

（4） 待ち行列に入っているデータの平均値を返す関数 `average()` を追加したい。以下の要件を満たす `average()` の関数定義、および、他の部分への必要な変更点を示せ。

- `average()` は値を `double` 型で返す。
- 待ち行列の操作のたびに全ての要素にアクセスすることは避ける。
- `average()` の呼び出しのたびに待ち行列の全ての要素にアクセスすることは避ける。
- 計算値のオーバーフローは考えなくてよい。

計算機理論 I

計算機理論 I を選択する場合には，解答用紙の指定欄に計算機理論 I と記入せよ．なお，計算機理論 I と計算機理論 II の両方を選択することはできない．

[1] 集合 E は次の (i),(ii),(iii),(iv) の系列（sequence）をその順に並べて得られる系列の集合であるとする．

(i) ‘+’ または ‘−’．ただしこれは省略してもよい．
(ii) ‘0’ と ‘1’ から構成される系列．(iv) の系列の長さが 1 以上ならば，その長さは 0 でもよい．
(iii) 小数点 ‘.’
(iv) ‘0’ と ‘1’ から構成される系列．(ii) の系列の長さが 1 以上ならば，その長さは 0 でもよい．

以下の問に答えよ．

(1) E を表現する正規表現（regular expression）を求めよ．
(2) E を受理する（accept）ε 動作付き非決定性有限オートマトン（nondeterministic finite automaton with ε moves）の状態遷移図（state transition diagram）を示せ．

[2] 以下の問では，「かつ」，「または」，「ならば」，「〜ではない」を表わす論理記号は，それぞれ，「$\wedge$」，「$\vee$」，「$\rightarrow$」，「$\neg$」を用いることとする．すなわち，「A かつ B」は「$A \wedge B$」，「A または B」は「$A \vee B$」，「A ならば B」は「$A \rightarrow B$」，「A ではない」は「$\neg A$」のように表わすものとする．

(1) C 言語の代入文（assignment）

```
x = 2*x + 3*y;
```

を実行した直後に次の論理式で表わされる条件が成り立っているとする．

$$x < y$$

この代入文を実行する直前に成り立っている条件を論理式として与えよ．

(2) C 言語の if 文

```
if (y < 0) x = -2*y; else x = 3*y;
```

を実行した直後に次の論理式で表わされる条件が成り立っているとする．

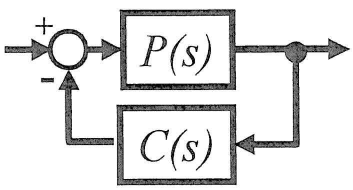
$$x < z$$

この if 文を実行する直前に成り立っている条件を論理式として与えよ．

計算機理論 II

計算機理論 II を選択する場合には，解答用紙の指定欄に計算機理論 II と記入せよ．なお，計算機理論 I と計算機理論 II の両方を選択することはできない．

[1] 下図及び下式において、$P(s)$は安定な制御対象、$C(s)$は制御器である。
この制御系について、以下の質問に答えよ。なお、解答に際して解の導出過程も明記すること。

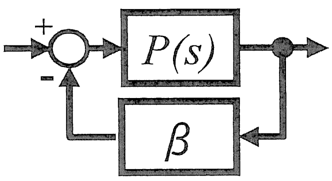


$$P(s)=\frac{3}{s^2+4s+3},\ C(s)=a+\frac{b}{s}$$

(1)開ループ一巡伝達関数$P(j\omega)C(j\omega)$の偏角が$-180°$となる周波数、すなわち位相交差周波数ω_{cp}と、その時の$P(j\omega_{cp})C(j\omega_{cp})$を求めよ。

(2) ω_{cp}と$P(j\omega_{cp})C(j\omega_{cp})$から安定限界となる時の$a,b$が満足すべき条件を求めよ。

[2] 下式の開ループおよび閉ループ利得で表されるフィードバック制御系について、以下の質問に答えよ。なお、解答に際して解の導出過程も明記すること。



制御対象 : $P(s)=\dfrac{\mu_0\omega_0^2}{(s+\omega_0)(ks+\omega_0)}$　$(\mu_0, k>0)$

$P(s)$の直流利得 : $P(0)=\mu_0$

フィードバック利得 : $\beta\ (>0)$

閉ループ系の伝達関数 : $G(s)=\dfrac{P(s)}{1+P(s)\beta}$

(1) 制御系の安定性と、$\mu_0\beta$の関係について述べよ。

(2) $k=100$の場合、位相余裕を45°とすると、$\mu_0\beta$は、いくらとなるか。なお、解を求める際、次式の近似を用いても良い。

$$\tan^{-1}x\approx 90° \quad (x\gg 1)$$

(3) Gの振幅特性$|G(j\omega)|$がピークを持たない条件を求めよ。

Translations of technical terms

制御対象:controlled object
制御器:controller
制御系:control system
開ループ一巡伝達関数:open loop transfer function
偏角:argument
位相交差周波数:phase crossover frequency
安定限界:stability limitation
ﾌｨｰﾄﾞﾊﾞｯｸ制御系:feedback control system
直流利得:D.C. gain
ﾌｨｰﾄﾞﾊﾞｯｸ利得:feedback gain
閉ループ系:closed loop system
安定性:stability
位相余裕:phase margin
振幅特性:gain characteristics

ハードウェア

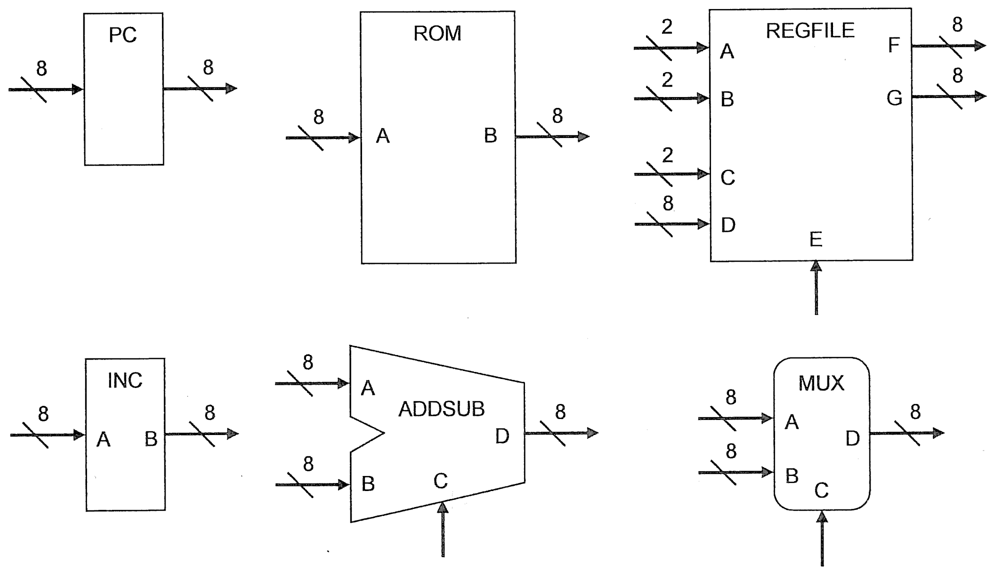
[1] 以下に述べる仕様を満たすプロセッサを設計し、回路図を描け。

- 加算命令、減算命令、ジャンプ命令の 3 種類の命令を有する。
- すべての命令は 1 クロック・サイクルで実行を完了する。
- 8 ビットのプログラム・カウンタを有する。
- 8 ビットのレジスタを 4 本備えるレジスタ・ファイルを有する。
- 1 つの命令の長さは 8 ビットであり、下の図に示すように、4 つのフィールド OP、RS、RT、RD に分けて使用する。フィールドの長さはすべて 2 ビットである。OP は命令操作コードであり、実行すべき命令の種類を表す。RS、RT、RD はオペランドとなるレジスタの番号を表す。

OP	RS	RT	RD

- OP が 00 のときは加算命令であり、RS と RT が示す 2 つのレジスタ間の加算を行い、その結果を RD が示すレジスタに書き込む。
- OP が 01 のときは減算命令であり、RS が示すレジスタの値から RT が示すレジスタの値を減じた結果を、RD が示すレジスタに書き込む。
- OP が 10 または 11 のときはジャンプ命令であり、RS が示すレジスタの値をプログラムカウンタに代入する。
- ジャンプ命令以外のときは、プログラム・カウンタの値は 1 増加する。

設計の際には、次に図示する部品と、2 入力 AND ゲート、2 入力 OR ゲート、NOT ゲートを必要な数だけ使用してよい。なお、必ずしもすべての種類の部品やゲートを使用しなくてもよい。



図において、信号線に与えられた数字は、その信号線のビット数を表す。数字が与えられていない信号線は 1 ビットである。また、クロック信号線は省略している。各部品の動作は以下の通りである。

- PC はプログラム・カウンタに用いるためのレジスタであり、クロック信号に同期して値が更新される。
- ROM は読み出し専用メモリであり、最大 256 命令からなるプログラムが格納される。入力 A が示すアドレスの命令を即座に B に出力する。
- REGFILE は 8 ビットのレジスタを 4 本備えるレジスタ・ファイルである。2 つのレジスタの読み出しと、1 つのレジスタの書き込みを、同一クロック・サイクルで行うことができる。A が示すレジスタの値を F に、B が示すレジスタの値を G に、それぞれ即座に出力する。また、E が 1 のとき、クロック信号に同期して、C が示すレジスタに D の値を書き込む。E が 0 のときは、レジスタへの書き込みは行われない。
- INC は加算器であり、A に 1 を加えた値を B に出力する。
- ADDSUB は加算と減算の 2 種類の演算を行うことができる演算器である。C が 0 のとき、A に B を加えた値を D に出力する。C が 1 のとき、A から B を減じた値を D に出力する。
- MUX はマルチプレクサである。C が 0 のとき、A の値を D に出力する。C が 1 のとき、B の値を D に出力する。

[2] 問題[1]におけるマルチプレクサ MUX は、1 ビットのマルチプレクサを 8 個並べることにより構成できる。1 ビットのマルチプレクサの回路図を、2 入力 AND ゲート、2 入力 OR ゲート、NOT ゲートを必要な数だけ用いて描け。なお、必ずしもすべての種類のゲートを使用しなくてもよい。

English translations of technical terms

プロセッサ	:	processor
回路図	:	circuit diagram (or circuit schematic)
プログラム・カウンタ	:	program counter
レジスタ	:	register
レジスタ・ファイル	:	register file
フィールド	:	field
命令操作コード	:	operation code (or opcode)
オペランド	:	operand
読み出し専用メモリ	:	read-only memory
マルチプレクサ	:	multiplexer

ソフトウェア

マージソート (merge sort) の 時間計算量(じかんけいさんりょう)(time complexity) を考える。ソート対象を 2 つに分割するコスト (cost) はソート対象の要素数に関係なく定数 a であるとする。2 つの要素の大小を比較し、大きい要素を出力するコストは定数 b であるとする。要素数が N の時のマージソートの時間計算量を $C_2(N)$ と表す。ただし、$C_2(1) = 1$ とする。

1) このとき $C_2(N)$ を $C_2(N/2)$、a、b、N で表せ。

2) $N = 2^k$ のとき、$C_2(N)$ を a、b、N のみで表せ。

3) $C_2(N)$ の N に対するオーダ (order) を示せ。

4) つぎに $N = 3^{k'}$ として、ソート対象を 3 分割してマージソートする場合のコスト $C_3(N)$ を a'、b'、N で表せ。ただし、ソート対象を 3 つに分割するコストはソート対象の要素数に関係なく定数 a' 、3 つの要素の大小を比較して最大の要素を出力するコストは定数 b' であるとする。

5) $C_2(N)$ と $C_3(N)$ の比較から、分割数 l を増加させていくと時間計算量 $C_l(N)$ は減少するので、l を要素数 N に応じて、増加させるようにすれば、より高速なアルゴリズムが構成できるように見える。しかし、実際にはそうなるとは限らない。この理由を示せ。