北海道大学　大学院情報科学研究科
情報理工学専攻 修士課程入学試験
平成 28 年 8 月 18 日 10:00-12:00

専門科目 1

受験上の注意

● 本冊子内の 5 問，問 1 （基礎数学），問 2 （情報数学），問 3 （確率・統計），問 4 （コンピュータ基礎工学），および問 5 （プログラミング）から 3 問を選択し解答すること．

● 選択問題チェック票に受験番号および，選択した科目に印を記入すること．

●すべての答案用紙に，受験番号，選択した問題番号(例えば，問 3 など)を必ず記入すること．

●答案用紙は 3 枚である．この他に下書き用の草案紙 3 枚を配付する．

●解答は，問題ごとに別々の答案用紙に記入すること(裏面を使用してもよい．答案用紙が不足したり，破損したりした場合には試験監督員に申し出て受け取ること)．

●解答が複数枚にわたる時は，1/2，2/2 のように答案用紙にページ番号を必ず付すこと，及び受験番号，選択した問題番号を各ページに記入すること．

●問題冊子，草案紙は持ち帰り，選択問題チェック票とすべての答案用紙とを提出すること．

●机の上に置いてよいものは，筆記用具（鉛筆，消しゴム，鉛筆削りなど），時計，および特に指示があったもののみである．時計は計時機能のみを使用し，アラームの使用を禁ずる．携帯電話、スマートフォン、タブレットコンピュータ等は電源を切ってかばんの中にしまうこと．電卓，電子辞書などは使用不可である．

問１．基礎数学

［１］実数 $x > 0$ の関数 $\Gamma(x)$ を次の式で定義する．

$$\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} du$$

だだし e は自然対数の底である．以下の問いに答えよ．

（１） $\Gamma(1)$ の値を求めよ．

（２） $\Gamma\left(\frac{1}{2}\right)$ の値を求めよ．ただし，$\int_0^\infty e^{-x^2} dx = \frac{1}{2}\sqrt{\pi}$ を用いてよい．

（３） 次の式が成り立つことを証明せよ．

$$\Gamma(x+1) = x\Gamma(x)$$

［２］ 以下の問いに答えよ．ただし，行列の右肩の T は元の行列の転置であることを示す．

（１） 任意の p 行 q 列の行列 A と q 行 r 列の行列 B に対して，$(AB)^{\mathrm{T}} = B^{\mathrm{T}} A^{\mathrm{T}}$ であることを証明せよ．

（２） A と B がともに p 行 p 列の対称行列のとき，AB は常に対称行列になるか否か，その理由とともに答えよ．

（３） 行列 A が逆行列を持つとする．λ が A の固有値であるとき，$\lambda \neq 0$ であること，および $\frac{1}{\lambda}$ は A の逆行列の固有値であることを証明せよ．

（４） 実対称行列 A の固有値が全て正であるとき，

$$B^2 = A$$

を満たす行列 B が存在することを証明せよ．

問 2. 情報数学

［1］ 命題論理とブール代数に関する以下の問いに答えよ．

（1） 以下の論理式(a)，(b)がトートロジー（恒真式）であるか否かを答えよ．ただし，$\wedge$ は論理積，$\vee$ は論理和，$\rightarrow$ は含意，$\sim$ は否定を表し，論理式 p, q に対して $p \rightarrow q$ は $\sim p \vee q$ と同値であり，$p \leftrightarrow q$ は $(p \rightarrow q) \wedge (q \rightarrow p)$ と同値である．

(a) $(p \rightarrow q) \vee \sim (p \leftrightarrow \sim q)$　　(b) $((p \rightarrow \sim q) \wedge (r \rightarrow q) \wedge r) \rightarrow (\sim p)$

（2） 三つの入力端子 A, B, C と出力端子 Y を有する論理回路において，出力 Y が入力 A, B, C の関数として $Y = ABC + A\bar{B}C + \bar{A}B$ と表現されるとする．ここで積は AND ゲート，和は OR ゲート，記号 $\bar{}$ は NOT ゲートに接続されていることを意味する．この論理回路にビット列 $A = 00001111$，$B = 00110011$，$C = 01010101$ が入力されたときの出力 Y のビット列を求めよ．

（3） ブール関数（ブール表現ともいう） $E(x, y, z, t) = y't' + y'z' + yzt' + x'y'zt$ を最簡形に変形せよ．ただし，記号 ' は補元を表す．また，ブール関数の最簡形とは，変数の積で表される項の和であって，項の数が最小である表現のうち，式に現れる変数の総数が最小のものとする．

［2］ 二項関係および有向グラフに関する以下の問いに答えよ．ただし，$V = \{3, 5, 7, 10, 15, 21\}$ とし，xRy を V 上の二項関係で「x が y より小さく，かつ x と y が互いに素である」と定義する．

（1） 二項関係 R を順序対の集合として記述せよ．

（2） 二項関係 R に対応する有向グラフを描け．ただし，V を頂点集合とし，関係 xRy は頂点 x から頂点 y への有向辺で表すものとする．

（3） （2）で得られた有向グラフにおける，各頂点の正の次数（出次数）と負の次数（入次数）を求めよ．

（4） （2）で得られた有向グラフの全域木（グラフを張る有向木）は全部でいくつあるか求めよ．また，その求め方を説明せよ．

問 3. 確率・統計

［1］b 個の黒い球と r 個の赤い球が入っている壺がある．この壺から無作為に 1 個の球を取り出し，それと同じ色の球を c 個付け加えて，壺に戻す．この操作を繰り返す．

（1）1 回目に取り出された球が赤である確率，および 1 回目と 2 回目の両方とも取り出された球が赤である確率を求めよ．

（2）2 回目に取り出された球が赤である確率を求めよ．

（3）2 回目に取り出された球が赤であることだけを知った場合，1 回目に取り出された球が赤である確率を求めよ．

［2］次に示す式が(同時)確率密度関数となる確率変数 (X,Y) に関して以下の問いに答えよ．

$$f(x,y)=\begin{cases} c & x^2+y^2<1 \text{のとき} \\ 0 & \text{その他の場合} \end{cases}$$

（1） $f(x,y)$ が確率密度関数であることから定数 c を求めよ．

（2） X の期待値 $\mathrm{E}(X)$ を求めよ．

（3） X と Y の共分散 $\mathrm{Cov}(X,Y)$ を求めよ．

（4） X と Y は独立でないことを証明せよ．

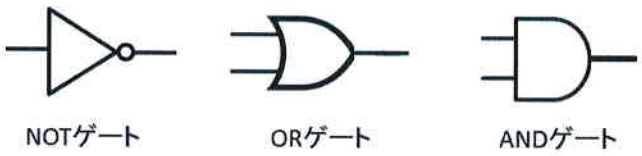
問 4. コンピュータ基礎工学

［1］補数に関する以下の問いに答えよ.

（1） コンピュータにおいて補数を用いる利点をあげよ.
（2） 負の整数を 2 の補数で表現するとき， n ビットの 2 進数で表現できる整数の範囲を，n を用いて示せ.
（3） 同様に負の整数を 1 の補数で表現するときの範囲を示せ.
（4） 1 の補数と 2 の補数の表現できる範囲の違いが生じる理由を記せ.
（5） 補数の考え方からすると，10 進数における 10 の補数，9 の補数を考えることが可能である．10 進数の 42−16 の計算を，10 の補数を用いて行った場合の計算過程を示せ．桁数は 3 ケタとする.

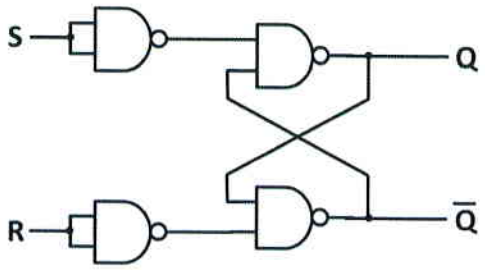
［2］組合せ回路，順序回路に関する以下の問いに答えよ.

（1） 以下の論理ゲートを用いて，論理式 $Q = A \cdot \bar{B} + B \cdot C$ を表す回路を構成せよ.
ここで，+は論理和 OR，・は論理積 AND，¯は否定 NOT を表す.
各ゲートは複数回用いてもよい.



（2） すべての論理式は NAND（否定論理積）ゲートで表現することができる．例えば，NOT 演算を NAND ゲートで以下のように表すことができる．これを用いて，AND 演算と OR 演算を実現する回路を NAND ゲートのみで表せ.

（3） 状態を記憶するための記憶回路として，フリップフロップがある．基本的なフリップフロップ回路であるRSフリップフロップは下図にあるようにNANDを用いて表される．このフリップフロップに対して，現在の出力Qの状態をQ_nとし，入力S, Rが与えられたときの次の安定出力状態Q_{n+1}を表す状態遷移表は以下のようになる．(a)～(f)の値を求めよ



NANDを用いたフリップフロップ

RS−フリップフロップの状態遷移表

入力		現在の状態	次の状態	
S(Set)	R(Reset)	Qn	Qn+1	
0	0	0	(a)	
0	0	1	(b)	
0	1	0	(c)	(リセット)
0	1	1	(d)	(リセット)
1	0	0	(e)	
1	0	1	(f)	
1	1	禁止		
1	1	禁止		

（4） このフリップフロップは$S=1, R=1$ を入力すると$Q=1, \bar{Q}=1$となり，その後$S=0$, $R=0$にするとQ，$\bar{Q}$ が定まらなくなるために，$S=1$，$R=1$ の入力は禁止される．これを$S=1$，$R=1$ が入力されたときにもリセット（(3)の状態遷移表参照）として働く回路としたい．元のRSフリップフロップを変更することによってこの論理回路を求め，それを図示せよ．

［3］異なるコンピュータ間でデータのやりとりを行う場合の手順やルールを定めたものは，通信プロトコルと呼ばれる．このプロトコルはISOが定めたOSI基本参照モデルによって，7つの階層に分けられ，各階層に必要な機能が定義されている．以下に示す層の機能，役割について端的に記述せよ．

（1） 物理層
（2） データリンク層
（3） ネットワーク層
（4） プレゼンテーション層

問 5. プログラミング

[1] その数自身を除く約数の和が，その数自身と等しい自然数を完全数と言う．例えば 6=1+2+3 や 28=1+2+4+7+14 が完全数である．次の C 言語プログラムの**ソースコード**の空欄を埋め 10000 以下のすべての完全数を出力するプログラムを完成させよ．

ソースコード

```
#include <stdio.h>

int divisor_sum(int n){
    int j, sum;
    sum = 0;
    for( (ア) ){
        if ( (イ) )
            sum += j;
    }
    return sum;
}

int main(){
    int i;
    for(i=1; i<=10000; i++){
        if (i== (ウ) )
            printf("%d¥n",i);
    }
    return 0;
}
```

[2] n 番目のフィボナッチ数 F_n は漸化式 $F_0 = 0, F_1 = 1, F_{n+2} = F_n + F_{n+1}$ $(n \geq 0)$ で定義される数であり，例えば F_{50}=12586269025 である．次の**ソースコード 1** は，50 番目までのフィボナッチ数を出力する C 言語プログラムである．このプログラムを実行したところ計算結果を得るまでに時間がかかってしまったため，これを改善したい．**ソースコード 2** の(ア)～(オ)を適切に埋め改善版プログラムを完成させると共に，効率が改善される理由を説明せよ．

ソースコード 1

```
#include <stdio.h>

long fib(int n){
    if(n==0){
        return 0;
    }else if(n==1){
        return 1;
    }else{
        return fib(n-1)+fib(n-2);
    }
}

int main(){
    int i;
    for(i=0; i<=50; i++){
        printf("fib(%d) = %ld¥n", i, fib(i));
    }
    return 0;
}
```

ソースコード 2

```
#include <stdio.h>

long fib(int n){
    long array[51];
    int i;
    array[0] = (ア);
    array[1] = (イ);
    for( (ウ) ){
        (エ);
    }
    return (オ);
}

int main(){
    int i;
    for(i=0; i<=50; i++){
        printf("fib(%d) = %ld¥n", i, fib(i));
    }
    return 0;
}
```

[3] 書籍の裏表紙や奥付ページには通常「ISBN978-4-…」から始まる下記の例のような ISBN（国際標準図書番号：International Standard Book Number）と呼ばれる文字列が印字されている．

ISBN978-4-949999-12-0
C3000 ¥2000E

9784949999120

（日本図書コード管理センター発行『ISBN コード／日本図書コード／書籍 JAN コード　利用の手引き　2010 年版』より）

現行規格は 13 桁の数字であり，このうち最後の桁の数字はチェックディジットと呼ばれる番号読み取りエラーを検出するための検査数字である．この検査数字は「モジュラス 10/ウェイト 3」と呼ばれる次の方式で算出される：チェックディジットを除いた最初 12 桁の数字の一番左側の桁から順に 1, 3, 1, 3, …を掛けてそれらの和を取り s とする．s を 10 で割った余り x を 10 から引いた値を結果とする．ただし x=0 なら 0 を結果とする．例えば，978-4-1234-5678-4 の場合，チェックディジット 4 は，s = 1×(9+8+1+3+5+7) + 3×(7+4+2+4+6+8) = 126，s÷10 = 12 余り 6 で x = 6，よって 10−x として得られる．次の**ソースコード**は入力された 13 桁の ISBN のチェックディジットを検査し，妥当な ISBN 番号かどうかを判定する C 言語プログラムであり，右はその実行画面の例である．**ソースコード**中の空欄(ア)(イ)を埋め，プログラムを完成させよ．入力は実行画面の例のようにスペース区切りで 0〜9 の数字が 13 個入力されるものとする．

ソースコード isbn_check.c

```
#include <stdio.h>

int check_digit(int code[]){
        (ア)
}

int main(){
    int i, v;
    int isbn[13];
    for(i=0; i<13; i++){
        scanf("%d", (イ) );
    }
    v = check_digit(isbn);
    if(isbn[12] == v)
        printf("Valid ISBN¥n");
    else
        printf("Invalid ISBN¥n");
    return 0;
}
```

実行画面

```
$ cc isbn_check.c -o check.exe
$ ./check.exe
9 7 8 4 9 4 9 9 9 9 1 2 0
Valid ISBN
$ ./check.exe
9 7 8 4 9 4 9 9 9 9 1 2 1
Invalid ISBN
$ ./check.exe
9 7 8 4 1 2 3 4 5 6 7 8 4
Valid ISBN
$ ./check.exe
9 7 8 4 1 2 3 4 5 6 7 8 5
Invalid ISBN
```