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 29 年 8 月 24 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．

専門科目１

選択問題チェック票

１．　受験番号を記入せよ．

２．　問１から問 5 のうち選択した３問について以下の表中に○を記入せよ．なお，選択した問題番号と答案用紙に記入した問題番号が一致していることを十分よく確かめること．

３．　本チェック票は答案用紙と一緒に提出すること．

受験番号	

問１　基礎数学		選択した３問に○を記入すること
問２　情報数学		
問３　確率・統計		
問４　コンピュータ基礎工学		
問５　プログラミング		

問 1. 基礎数学

［1］$I_n = \int_0^{\pi/2} \sin^n x \, dx$ とする．以下の問いに答えよ．ただし，n は非負の整数である．

（1）I_0, I_1 を求めよ．

（2）I_n についての漸化式を示せ．

（3）I_n を求めよ．

（4）$J_k = \dfrac{1}{I_{k-1} I_k}$ (ただし，$k \geq 1$) とする．$\displaystyle\lim_{n\to\infty}\sum_{k=1}^{n}\frac{n}{n^2 + J_k^2}$ を求めよ．

［2］異なる n 次正方行列 A, B に対し，n 次正則行列 P により $P^{-1} A P$ および $P^{-1} B P$ が対角行列となるとき，A, B は P により同時対角化可能である，と言う．以下の問いに答えよ．

（1）A，B が P により同時対角化可能であるとき，$AB = BA$ であることを証明せよ．

（2）A の固有値がすべて異なり，かつ $AB = BA$ のとき，A, B は P により同時対角化可能であることを証明せよ．

問 2. 情報数学

［１］二項関係に関する以下の問いに答えよ．

集合 A 上の二項関係 R は以下の３つの性質を満たすとき，同値関係であるという．

(反射律)　任意の $x \in \mathrm{A}$ に対して，$x\mathrm{R}x$ が成り立つ．

(対称律)　任意の $x,y \in \mathrm{A}$ に対して，$x\mathrm{R}y$ ならば $y\mathrm{R}x$ が成り立つ．

(推移律)　任意の $x,y,z \in \mathrm{A}$ に対して，$x\mathrm{R}y$ かつ $y\mathrm{R}z$ ならば $x\mathrm{R}z$ が成り立つ．

（１）自然数 x,y に対し，

$$x\mathrm{R}y \Leftrightarrow |x - y|=6$$

と定義すれば R は自然数の集合上の二項関係となる．このとき R は，反射律，対称律，推移律を満たすか．各々の性質について，満たす場合は証明し，満たさない場合は反例をあげよ．

（２）（１）の二項関係 R を含む最小の同値関係を求めよ．

［２］ブール関数に関する以下の問いに答えよ．

ブール関数 $f(x,y,z,w)=(x+y+z')(x+z'+w)(y'+z)$ を考える．ただし，任意のブール変数 x, y に対し $x+y$, xy はそれぞれ x と y のブール和，ブール積を，また x' は x の否定を表すものとする．

（１）$f(x,y,z,w)$ の否定 $f'(x,y,z,w)$ を変数の積で表される項の和で表せ．

（２）$f'(x,y,z,w)=1$ となる (x,y,z,w) の組をすべて求めよ．

（３）$f(x,y,z,w)$ の最簡形を求めよ．ただし，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．

［３］命題論理に関する以下の問いに答えよ．

左右２つに分かれている道があり，一方は A 村へ，もう一方は B 村に繋がっている．A 村，B 村どちらかの村人 x が，その分岐点に立っている．

（１）以下の３つの命題 P，Q，R を考える．

P:「A 村は右の道に繋がっている．」

Q:「村人 x は A 村の住民である．」

R:「村人 x は，右の道に繋がっている村の住民である．」

命題 P と論理的に同値な命題を，命題 Q と命題 R から作られた合成命題（論理和∨，論理積∧，否定∼を使って合成された命題）により表せ．

（２）村人 x 自身に命題 R の真偽を答えてもらう．ただし，A 村の住民は常に正しい答えを言い，B 村の住民は常に間違った答えを言うものとする．このとき，村人 x の答えから A 村が右左どちらの道に繋がっているかを知ることができる理由を説明せよ．

問 3. 確率・統計

［1］ ある小学校の 1 年生 n 人 $(1 < n < 365)$ は，生まれた年がうるう年でなく，双子やそれ以上の多胎児でないことがわかっている．このとき，以下の問いに答えよ．なお，特に断りのない限り，n 人は独立に等確率で誕生したものとする．

（1） n 人の中に，同じ日に誕生した人が存在する確率 p_1 を，n を用いた式であらわせ．

（2） この学年に転校生が 1 名やってきたとき，転校生と同じ誕生日の人が n 人の中にいる確率 p_2 を，n を用いた式であらわせ．

［2］ 以下の各項で示す 2 つの値について，常に等しい場合はその根拠を，等しくない場合があればその具体例を，それぞれ示せ．

（1） 1 変量標本 $\{x_i\}$ $(i = 1,2,\ldots,n)$の，算術平均と中央値

（2） 2 変量標本$\{(x_i, y_i)\}$ $(i = 1,2,\ldots,n)$の相関係数 r と，
x_iに実定数 a $(a \neq 0)$ を乗じて置き換えた 2 変量標本 $\{(ax_i, y_i)\}(i = 1,2,\ldots,n)$の相関係数 r_a

（3） 確率関数が $\mathrm{P}(X = x) = e^{-\lambda}\frac{\lambda^x}{x!}$ $(x = 0,1,2,\ldots)$ で与えられるポアソン分布の，期待値と分散

［3］ 累積分布関数が以下に示す$F(x)$で与えられる連続一様分布の歪度と尖度を求めよ．ただし，正規分布の尖度は 3 とする．

$$F(x) = \begin{cases} 0 & (x < -0.5) \\ x + 0.5 & (|x| \leq 0.5) \\ 1 & (x > 0.5) \end{cases}$$

問 4. コンピュータ基礎工学

［1］コンピュータの数値表現に関する以下の問いに答えよ.

（1） 10 進数の 97 を 2 進数で表せ.
（2） 10 進数の −43 を 2 進数 7 桁の 2 の補数表現，1 の補数表現でそれぞれ表せ.
（3） 10 進数の 97−43 を 2 進数 7 桁の 2 の補数表現で計算した場合の計算過程と結果を示せ.
（4） 数値表現の浮動小数点方式における正規化とは何か簡潔に記述せよ.
（5） ネットワークバイトオーダ（ビッグエンディアン）で複数バイトを送信する場合の規則を簡潔に記述し，16 進数の 8532AEC7 を送信する具体的な順序を示せ.

［2］計算機アーキテクチャに関する下記の問いに答えよ.

（1） 多くの計算機アーキテクチャでは，プログラム内蔵方式を採用し，プログラムカウンタ（PC）をプロセッサ内部に有している．プログラムカウンタとは何か，またプログラム内蔵方式において，プログラムカウンタが何故必要であるか簡潔に答えよ.
（2） 図 1 に示すダイレクトマップ方式のキャッシュ機構を仮定した場合に，2 進数 110000，001000，001100 のメモリアドレスに対して順にロード命令を実行したときの最終的なキャッシュテーブルの状態を図示せよ．なお，メモリアドレスは 6bit のバイトアドレス方式とし，図 1 のキャッシュテーブルの状態及び図 2 のメモリの内容から処理を開始するものとする.

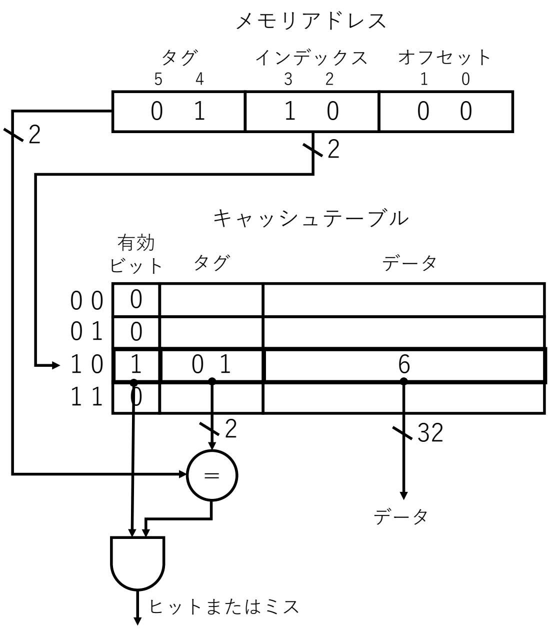

図 1. キャッシュ機構

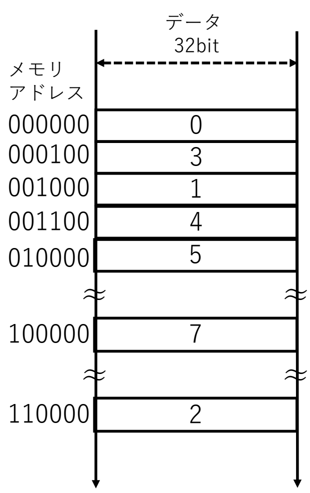

図 2. メモリアドレス空間

（問題は裏面に続く）

［３］オペレーティングシステムに関する下記の問いに答えよ．

（１） 図 3 はプロセススケジューリングの例を示しており，プロセスの到着を矢印，プロセス処理の継続を四角で表現している．図 3 に示す例を参考に，表 1 に示す 4 つのプロセスを単一プロセッサコアで処理する場合の以下の 2 つのアルゴリズムのスケジューリング結果をそれぞれ図示せよ．なお，プロセス切り替えに要するオーバヘッドは無視してよいものとする．

(a) 到着順（FCFS: First Come First Served）

(b) ラウンドロビン（タイムクォンタムを 2 とする）

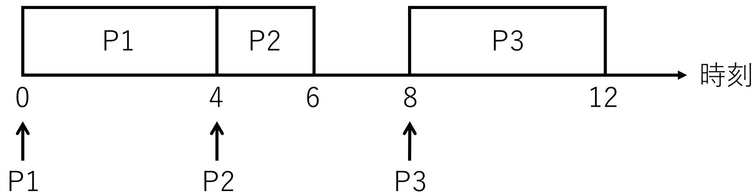

図 3．プロセススケジューリングの例

表 1．プロセスの到着

プロセス	到着時刻	処理時間
P1	0	6
P2	3	2
P3	7	4
P4	9	2

（２） ページング方式を採用する仮想記憶において，ページフォールトとは何か簡潔に記述せよ．また，ページフォールトが発生した場合，オペレーティングシステムはどのような処理を行うか簡潔に記述せよ．

問 5. 計算機プログラミング

[１]　一般にn個の中からk個を選ぶ組み合わせの数を $\binom{n}{k}$ と書き，次の式で定義される.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

この式をこのまま計算機上で実装した場合，比較的小さなnに対しても$n!$でオーバーフローする危険性がある．例えば：

$$\binom{13}{5} = \frac{13!}{5! \cdot 8!} = \frac{6{,}227{,}020{,}800}{120 \cdot 40{,}320} = 1{,}287$$

となり，最終結果はオーバーフローしない値でも，C 言語の 32 ビットの int 型ならば 13!のところでオーバーフローしてしまう.

ここで，$\binom{n}{k}$ は以下の漸化式を用いて表現することもできる.

$$\begin{cases} \binom{n}{k} = \frac{n-k+1}{k}\binom{n}{k-1} & (k \geq 1) \\ \binom{n}{0} = 1 & (k = 0) \end{cases}$$

この計算は元々の式よりは大きな値を扱うことができ，実用的である．この計算は以下の C 言語のプログラムで実装できる．ソースコード中の（ア）～（イ）を適切に埋めてプログラムを完成させよ.

ソースコード 1-1

```
#include <stdio.h>
long combination(int n, int k) {
        int i;
        long c=1;
        for(i = 1;[  （ア）  ];i++) {
                c=[  （イ）  ];
        }
        return c;
}
int main() {
        int n, k;
        scanf("%d", &n);
        scanf("%d", &k);
        printf("combination(%d,%d) = %ld¥n", n, k, combination(n,k));
        return 0;
}
```

実行結果 1-1

```
13 ⏎ // 入力
5 ⏎ // 入力
combination(13,5) = 1287
```

更に $\binom{n}{k}$ は以下のように定義でき，これは再帰関数を用いて実装できる.

$$\begin{cases} \binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} & (n-1 \geq k \geq 1) \\ \binom{n}{k} = \binom{n}{0} = 1 & (k = 0\ Or\ k = n) \end{cases}$$

以下の C 言語プログラムのソースコード中の（ウ）～（エ）を適切に埋めてプログラムを完成させよ.

ソースコード 1-2

```
#include <stdio.h>
long combination(int n, int k) {
        if(      (ウ)      ) {
                return 1;
        } else {
                return (              (エ)              );
        }
}
int main() {
        ＜ソースコード 1-1 と同じため省略＞
}
```

＜実行結果 1-2 は実行結果 1-1 と同じため省略＞

[２]　一般的な C 言語の環境では string.h において文字列を扱う関数群が用意されている. その中に strtok という関数があり, これにより文字列を区切り文字(delimiter)で分割することができる. 使用方法は以下の通りである.

ソースコード 2-1

```
#include        <stdio.h>
#include        <string.h>
int main(void)
{
        char str[] = "One,Two,Three,Four";
        char *ptr;
        /* コンマ "," を区切り文字として使用 */
        ptr = strtok(str, ",");
        while ( ptr != NULL) {
                printf("%s¥n", ptr);
                ptr = strtok(NULL, ",");
        }
        return 0;
}
```

実行結果 2-1

```
One
Two
Three
Four
```

このように，strtok を利用するためには一手間必要であり十分に便利ではない．このため文字列を strtok と同様に区切り文字で分割する関数 split を実装することとした．以下の C 言語プログラム中の（ア）～（オ）を適切に埋めてプログラムを完成させよ．また，今回の実行結果には影響がないが，リストの最大長である MAX_LENGTH を考慮する実装とせよ．もしリストの長さが MAX_LENGTH を超える場合には，分割されずに残った文字列についてはリストの最後の要素に含まれていれば良い．

ソースコード 2-2

```
#include <stdio.h>
#define MAX_LENGTH 20

/* delimiter で文字列 str を破壊的に分割し，その分割された文字列のリストを作成する． */
/* 最後にリストの長さを返す． */
int split(char *str, char *list[], const char delimiter) {
        int index = 0;
        list[index++] = [  (ア)  ];
        while([          (イ)          ]) {
                str++;
                if([      (ウ)      ]) {
                        *str = '¥0';
                        [      (エ)      ]
                }
        }
        return index;
}

int main() {
        char str[] = "One,Two,Three,Four";
        char *result[MAX_LENGTH];
        int length;
        length = split(str, [  (オ)  ], ',');
        for(i = 0;i < length;i++) {
                printf("%s¥n", result[i]);
        }
        return 0;
}
```

＜実行結果 2-2 は実行結果 2-1 と同じため省略＞