平 成 １ ６ 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
第２次入学試験問題

<u>専　　　　門</u>

平成１６年２月１２日（木）
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、日本語から母国語への辞書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（<u>合計４科目を選択</u>して解答せよ）。
A群：次の３科目から<u>２科目を選択</u>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<u>２科目を選択</u>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。

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

解の導出過程を書くこと

[1] 次の不定積分を求めることを考える。

$$\int\sqrt{\frac{x-1}{x+1}}dx$$

以下の問いに答えよ。

(1) $t=\sqrt{\dfrac{x-1}{x+1}}$ とし、$\displaystyle\int\sqrt{\frac{x-1}{x+1}}dx$ を t に関する積分として表せ。

(2) (1)で求めた t に関する被積分関数を部分分数に展開せよ。

(3) 不定積分 $\displaystyle\int\sqrt{\frac{x-1}{x+1}}dx$ を求めよ。

[2] n 行 n 列の行列 $\mathbf{A}$ を考える。$\mathbf{A}^T\mathbf{A}$ は

$$\mathbf{V}^T(\mathbf{A}^T\mathbf{A})\mathbf{V}=\mathbf{Q}$$

のように対角化可能であることが知られている。ここで、$\mathbf{Q}$ は $\mathbf{A}^T\mathbf{A}$ の固有値

$\lambda_1,\lambda_2,\lambda_3,\cdots,\lambda_n$ を用いて、

$$\mathbf{Q}=diag\left(\lambda_1,\lambda_2,\lambda_3,\cdots,\lambda_n\right)\ (\lambda_1\geq\lambda_2\geq\lambda_3\geq\cdots\geq\lambda_n)$$

と表され、$\mathbf{Q}=diag\left(\lambda_1,\lambda_2,\lambda_3,\cdots,\lambda_n\right)$ は $\lambda_1,\lambda_2,\lambda_3,\cdots,\lambda_n$ を対角成分に持つ対角行列を示す。また、$\mathbf{V}$ は直交行列、$\mathbf{A}^T$ は行列 $\mathbf{A}$ の転置を表す。以下の問いに答えよ。

(1) 行列 $\mathbf{W}$ を $\mathbf{W}=diag(\sqrt{\lambda_1},\sqrt{\lambda_2},\sqrt{\lambda_3},\cdots,\sqrt{\lambda_n})$ とするとき、行列 $\mathbf{U}=\mathbf{A}\mathbf{V}\mathbf{W}^{-1}$ は直交行列であることを示せ。

(2) $\mathbf{A}=\mathbf{U}\mathbf{W}\mathbf{V}^T$ と書けることを示せ。

(3) 行列 $\mathbf{A}$ を $\mathbf{A}=\begin{pmatrix}1&-2\\2&-1\end{pmatrix}$ とする。行列 $\mathbf{U}$, $\mathbf{W}$, $\mathbf{V}$ をそれぞれ求めよ。

注)

不定積分	Indefinite integral	対角化	Diagonalization
直交行列	Orthogonal matrix		

確率・統計

（解の導出過程を書くこと）

［１］ さいころを２回振り，出た数値の差の絶対値を確率変数Xとする．Xの期待値$E[X]$を求めよ．

［２］ 確率変数X,Yが独立で，次の確率密度関数に従うとする．但し$\lambda>0$とする．

$$f_X(x)=\begin{cases}\lambda e^{-\lambda x} & (x\geq 0)\\ 0 & (x<0)\end{cases},\qquad f_Y(y)=\begin{cases}\lambda e^{-\lambda y} & (y\geq 0)\\ 0 & (y<0)\end{cases}$$

(1) X,Yの同時確率密度関数$g_{X,Y}(x,y)$を求めよ．

(2) $X+Y\leq z$となる確率$P(X+Y\leq z)$を求めよ．

(3) 定数θに対し$M(\theta)=E[e^{\theta X}]$により定義される関数を確率変数$X$のモーメント母関数と呼ぶ．確率密度関数$f_X(x)$に対する$X$のモーメント母関数（$\theta<\lambda$の場合のみ考えよ）を求めよ・

(4) 一般に，$E[X]=M'(0)$，$E[X^2]=M''(0)$となることを示せ．ここで$M'(\theta)$，$M''(\theta)$は，それぞれ$M(\theta)$の一階微分，二階微分を表す．

(5) 上記の$f_X(x)$に従う確率変数Xの分散$Var[X]$を，モーメント母関数を用いて計算せよ．

（上記の計算で$e^a=\sum_{n=0}^{\infty}\frac{a^n}{n!}$であることを利用しても良い　）

【専門用語の英訳】

確率変数: random variable, 期待値: expectation, 独立: independence,

確率密度関数: probability density function, 分散: variance,

同時確率密度関数: joint probability density function, 母関数: generating function

プログラミング

[I]　次に示すプログラムは，二分木（binary tree）を用いて，配列 data に格納された数値を小さい順に出力する C 言語によるプログラムである．ここで，NULL は何も指していないことを示す特殊なポインタ，関数 malloc はパラメータで指定されるサイズのメモリ領域を動的に割り付け，割り付けたメモリ領域の先頭番地を返す関数である．この時，メモリ領域の不足は発生しないものとする．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:    #include <stdlib.h>
 3:
 4:    #define N 5
 5:    int data[N] = { 5, 10, 3, 12 ,8 };
 6:
 7:    typedef struct tree {
 8:                int         val;
 9:                struct tree *left;
10:                struct tree *right;
11:            } TREE;
12:
13:    TREE *insert(TREE *t, int x)
14:    {
15:        if (t == NULL) {
16:            t = malloc(sizeof(TREE));
17:            t->val = x;
18:            t->left = NULL;
19:            t->right = NULL;
20:        }
21:        else if (x > t->val) {
22:            t->left = insert(t->left, x);
23:        }
24:        else {
25:            t->right = insert(t->right, x);
26:        }
27:        return(t);
28:    }
29:
30:    void traverse(TREE *t)
31:    {
32:        if (t != NULL) {
33:            traverse(t->right);
34:            printf("%d\n", t->val);
35:            traverse(t->left);
36:        }
37:    }
38:
39:    main()
40:    {
41:        TREE *t;
42:        int  i;
43:
44:        t = NULL;
45:        i = 0;
46:        while (i < N) {
47:            t = insert(t, data[i]);
48:            i = i + 1;
49:        }
50:        traverse(t);
51:    }
```

(1) このプログラムを実行した時，関数 insert は合計で何回呼び出されるか．また，このプログラムによって作られる二分木を図示せよ．

(2) このプログラムを，配列 data に格納された数値を大きい順に出力するように改造したい．関数 insert を修正する方法と，関数 traverse を修正する方法の2つの方法を示せ．プログラムの修正箇所が少ない場合には，元のプログラムとの差分のみを示してもよい．

(3) このプログラムは，関数 malloc によって割り付けられたメモリ領域を，プログラムが終了するまで解放しない．これを，関数 malloc によって割り付けられたメモリ領域を，プログラム中ですべて解放するように修正せよ．プログラムの修正箇所が少ない場合には，元のプログラムとの差分のみを示してもよい．なお，関数 malloc によって割り付けられたメモリ領域は，メモリ領域の先頭番地をパラメータとして関数 free を呼び出すことによって解放されるものとする．

(4) プログラミング言語によっては，関数 free（またはそれと同等の関数）を明示的に呼び出さなくても，参照できなくなったメモリ領域が自動的に解放される．そのための機構の名称を答え，そのような機構を持ったプログラミング言語を一つ挙げよ．

(5) 定数 N の値が一定という条件下で，配列 data にどのような入力値を与えた場合に，このプログラムの実行時間が最も長くなると予想されるか．

計算機理論

以下の問いに答えよ．なお，商 (quotient) と剰余 (remainder) を求める演算をそれぞれ div と mod で記し，最大公約数 (greatest common divisor) を求める演算を gcd で記す．

[1] 自然数 a と b から定まる数列 r_i を次のように与える．

$$\begin{cases} r_0 &= a \\ r_1 &= b \\ r_{i+2} &= \begin{cases} r_i \bmod r_{i+1} & (r_{i+1} \neq 0 \text{ の場合}) \\ 0 & (r_{i+1} = 0 \text{ の場合}) \end{cases} \end{cases}$$

また，N を $r_{N+1} = 0$ となる最小の自然数とする (mod の定義より $r_{i+1} \neq 0$ のときはいつでも $r_{i+1} > r_{i+2}$ となることから，N の存在は保証されている)．

このとき，自然数の集合 A_i を以下で定義すると，任意の i $(< N)$ に対し，$A_i = A_{i+1}$ が成立することを証明せよ．

$$A_i = \{d \mid d \text{ は } r_i \text{ と } r_{i+1} \text{ の公約数 (common divisor)}\}$$

[2] 数列 r_i を用いて，数列 q_i, x_i, y_i を次のように与える．このとき，任意の i $(\leq N)$ に対し，$ax_i + by_i = r_i$ が成立することを i に関する帰納法で証明せよ．

$$q_i = \begin{cases} r_i \text{ div } r_{i+1} & (i < N \text{ の場合}) \\ 0 & (i \geq N \text{ の場合}) \end{cases}$$

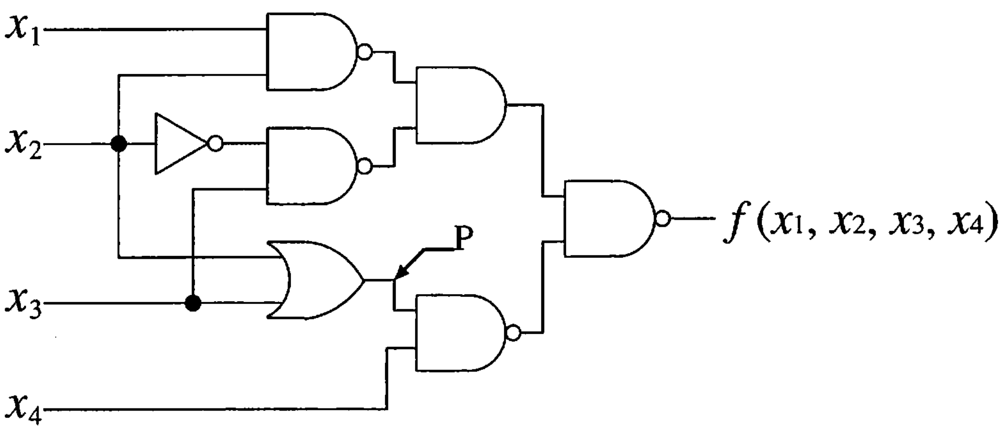
$$\begin{cases} x_0 &= 1 \\ x_1 &= 0 \\ x_{i+2} &= x_{i+1} - q_i x_i \end{cases} \qquad \begin{cases} y_0 &= 0 \\ y_1 &= 1 \\ y_{i+2} &= y_{i+1} - q_i y_i \end{cases}$$

[3] [1] の結果を用いて，任意の i $(< N)$ に対し $\gcd(r_i, r_{i+1}) = \gcd(r_{i+1}, r_{i+2})$ となることを示すことによって，次の関係が成立することを導け．

$$ax_N + by_N = \gcd(a, b)$$

ハードウェア

[1] 下図に示す組合せ論理回路 (combinational circuit) について，以下の問に答えよ。



1) 回路が計算する４変数論理関数 (4-variable logic function) $f(x_1,x_2,x_3,x_4)$ を表すカルノー図 (Karnaugh map) を示せ。
2) f の最小積和形表現 (minimum sum-of-products form) を求めよ。
3) 回路の P 点（OR ゲートの出力線）の１縮退故障 (stuck-at-1 fault) を検出する入力パターン (input pattern) を求めよ。

[2] プロセッサの命令パイプライン方式 (instruction pipelining) について説明せよ。

ソフトウェア

文脈自由文法 (context-free grammar) G=(T,N,P,S) について以下の問いに答えよ．ここで，終端記号集合 (terminal symbols)$T = \{a, (, :,)\}$，非終端記号集合 (non-terminal symbols)$N = \{S, L\}$ とし，生成規則集合 (production rules)P は以下の 4 つの規則からなるとする．

$$
\begin{array}{ll}
S \to (L) & \qquad S \to a \\
L \to L : S & \qquad L \to S
\end{array}
$$

[1] 以下の文 (sentence) の構文木 (parse tree) を示せ．

(i) $(a : a)$

(ii) $(a : (a : a))$

(iii) $(a : ((a : a) : (a : a)))$

[2] [1] における各文 (i)〜(iii) に対して最左導出 (leftmost derivation) を示せ．

[3] [1] における各文 (i)〜(iii) に対して最右導出 (rightmost derivation) を示せ．

[4] G についての演算子順位関係 (operator precedence relation) を以下の表に示す．この順位表を用いて $(a : (a : a))$ を構文解析する．ここで，\$ は入力開始と入力終端を表わす特別な記号であるとする．

	a	(	)	:	\$
a			$\gg$	$\gg$	$\gg$
(	$\ll$	$\ll$	$==$	$\ll$	
)			$\gg$	$\gg$	$\gg$
:	$\ll$	$\ll$	$\gg$	$\gg$	
\$	$\ll$	$\ll$			

以下に導出の最初の段階を示す．導出は右文形式 (right sentential form)，非終端記号を除いた記号列，ハンドル (handle) を $\langle$ と $\rangle$ で囲んだ 3 項組の系列で表すことにする．

$$
\begin{array}{ll}
 & \langle \$(a : (a : a))\$ \ ,\ \$ \ll (\ll a \gg : \ll (\ll a \gg : \ll a \gg) \gg) \gg \$ \ ,\ a \rangle \\
\to & \langle \$(S : (a : a))\$ \ ,\ \$ \ll (\ll : \ll (\ll a \gg : \ll a \gg) \gg) \gg \$ \ ,\ a \rangle
\end{array}
$$

以下，受理までの解析系列を示せ．