

筆答専門試験科目（午前）

2022 大修

情報工学系

時間 9:30~12:30

注 意 事 項

1. 各答案用紙の受験番号欄に受験番号を記入し、試験科目名欄に「情報工学系」と記入せよ。
 2. 次の5題のうち、1番～3番の3題は必ず解答し、4番と5番からは1題を選び解答すること。
 3. 解答は1題ごとに別々の答案用紙に、問題番号を明記した上で記入せよ。必要であれば、答案用紙の裏面に記入して良いが、答案用紙の表面にその旨を明記すること。
 4. 1枚の答案用紙に2題以上の解答を記入した場合はそれらの解答を無効とすることがある。
 5. 1題の解答を2枚以上の答案用紙に記入した場合はその解答を無効とすることがある。
 6. 電子式卓上計算機等の使用は認めない。
-

1. 3×3 実行列 A を

$$A = \frac{1}{3} \begin{pmatrix} 1 & -2 & -2 \\ -2 & 1 & -2 \\ -2 & -2 & 1 \end{pmatrix}$$

によって定義する。以下の間に答えよ。

- 1) 行列 A の行列式を求めよ。
- 2) I を 3×3 単位行列、 x を変数としたとき、行列 $xI - A$ の行列式を求めよ。
- 3) 行列 A の固有値を全て求めよ。
- 4) 行列 A の各々の固有値について、その固有値に対する固有空間の基底を求めよ。
- 5) ユニタリ行列の定義を述べ、その定義を行列 A が満たしていることを示せ。
- 6) エルミート行列の定義を述べ、その定義を行列 A が満たしていることを示せ。
- 7) $A = BDB^*$ を満たす行列 B と対角行列 D を求めよ。ただし B^* は B の共役転置である。

2. 以下の間に答えよ。

- 1) アルファベット $\{a, b, c\}$ 上の言語 $L = \{a^n b^m c b^m a^n \mid n \geq 1, m \geq 0\}$ を考える。 L は $abcbba$ や aca などの回文の集合である。生成規則集合 $\{S \rightarrow aSa, S \rightarrow aTa\}$ にさらに生成規則を 2つ付け加えて L を生成する文脈自由文法を完成させたい。ただし S は開始記号であり、 T は非終端記号である。付け加えるべき 2つの生成規則を書け。
- 2) 式 $(a+b)*((c/d)-e)$ を表す 2 分木 T が以下の図のように与えられている。ただし a, b, c, d, e は数値を表す記号であり、 $+, -, *, /$ は 2 項算術演算を表す記号である。

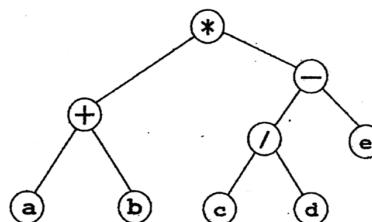


図2.1 2分木 T

- a) T を前順序（プレオーダー）でたどる順序を記せ。
- b) T を後順序（ポストオーダー）でたどる順序を記せ。
- 3) 命題変数またはその否定をリテラルという。命題論理式がリテラルの論理積の論理和であるとき、その命題論理式は論理和形であるという。 $\neg((A \vee B) \wedge (\neg(A \vee \neg C)))$ を論理和形に直せ。

(次ページにつづく)

(前ページのつづき)

4) $x < y$ を 自然数 \mathbb{N} の上の大小関係 (x は y より小さい) を表す述語とする。

- a) 「任意の x に対しそれより大きい y が存在する」を述語論理式に翻訳せよ。
- b) 述語論理式 $\forall x \forall y \{ \neg(x < y) \Rightarrow (y < x) \}$ の \mathbb{N} での真偽を理由とともに述べよ。
- c) 述語論理式 $\neg \forall x \forall y \{ \neg(x < y) \Rightarrow (y < x) \}$ の冠頭標準形を書け。

3. 以下のような条件 (A), (B) を満たす 2 分木を考える.

- (A) 子節点が存在する全ての節点について、その節点の値が子節点の値より大きいか等しい.
(B) 一番深いレベル以外は節点が全て詰まっており、一番深いレベルでは節点が左に詰まっている。

このような木を以下ではヒープ木と呼ぶ。ヒープ木を幅優先で左から走査しながら節点の値を順番に格納した配列を以下ではヒープ配列と呼ぶ。配列の添字は 0 から始まるものとし、 n 個の要素からなる配列 b を $b[0..n-1]$ と表記する。図 3.1 は、ヒープ木の例とそれに対応するヒープ配列 $c[0..9]$ である。

- 1) $b[0..n-1]$ をヒープ配列として、それに対応するヒープ木を T とする。
 - a) T の高さを h とするとき、 n の最大値を h で表せ。ここで木の高さとは、根節点と一番深いレベルの節点間にある枝の数とする。例えば、図 3.1 のヒープ木は高さ 3 である。
 - b) T において節点 p の左の子節点を q 、右の子節点を r とし、 p, q, r がそれぞれ $b[s], b[t], b[u]$ に対応するとき、 t と u を s で表せ。
 - c) $b[0..n-1]$ における n 個の要素が相異なる値を持ち、その最大値と最小値を持つ要素をそれぞれ $b[v], b[w]$ とするとき、 v と w それぞれの値もしくは値の範囲を答えよ。
- 2) ヒープ配列からは最大の要素を効率良く取り出すことができるるので、最大の要素を取り出し、それを除いた配列をヒープ配列として構成する処理を繰り返すことによって、要素を大きい順に次々と取り出すことができる。この考えを応用したデータ整列のアルゴリズムはヒープソートと呼ばれる。図 3.2 はヒープソートを実装した C 言語の関数である。

- a) `func1` は上記の下線部を実行する関数である。以下の説明を読んで、図 3.2 の空欄 [ア], [イ], [ウ] を埋めよ。

最大要素の取り出しとヒープ配列の構成について、図 3.1 を用いて説明する。まず、 $c[0]$ と $c[9]$ の値を交換して最大の要素を $c[9]$ に格納する。これは、図 3.1 のヒープ木において根節点の「12」と葉節点の「3」を交換することに対応する。次に、 $c[9]$ を除く $c[0..8]$ をヒープ配列として構成する。この処理は `func1(c, 0, 8)` を呼び出すことで実行される。図 3.3 はこの処理を木構造で示しており、(i), (ii), (iii) の各段階において値が「3」に更新された節点を x と呼ぶ。図 3.3(i) では、 x とその子節点以外の親子は条件 (A) を満たしているので、 x とその子節点が条件 (A) を満たすかどうか調べて、満たさない場合は x と子節点の値を交換する。ここで、 x と値を交換するのは、2 つ存在する子節点のうち値が大きい方だけである。図 3.3(ii) でも (i) と同様の処理を行う。 x とその子節点が条件 (A) を満たすか、図 3.3(iii) のように x に子節点が存在しなければ処理を終了する。

- b) `func1` は、図 3.3 のように x とその子節点以外の親子が条件 (A) を満たすことを前提としている。`func2` は、そのような前提が成立しない場合でも `func1` を使用して、与えられた配列をヒープ配列として構成する関数である。`func2` の第 1 引数と第 2 引数には、それぞれ配列とその要素数を与える。図 3.2 の空欄 [エ], [オ], [カ] を埋めよ。
- c) `func3` は、与えられた配列をヒープソートによって昇順に整列する関数である。`func3` の第 1 引数と第 2 引数には、それぞれ配列とその要素数を与える。図 3.2 の関数を可能な限り使用して、図 3.2 の空欄 [キ], [ク], [ケ] を埋めよ。

(次ページにつづく)

(前ページのつづき)

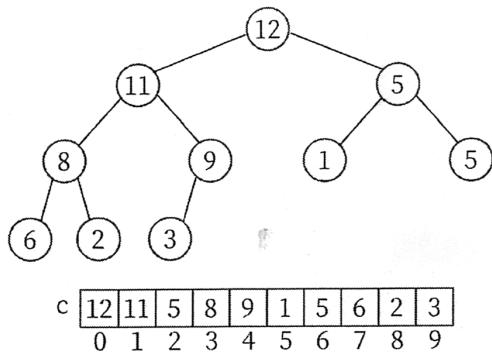


図 3.1: ヒープ木とヒープ配列の例

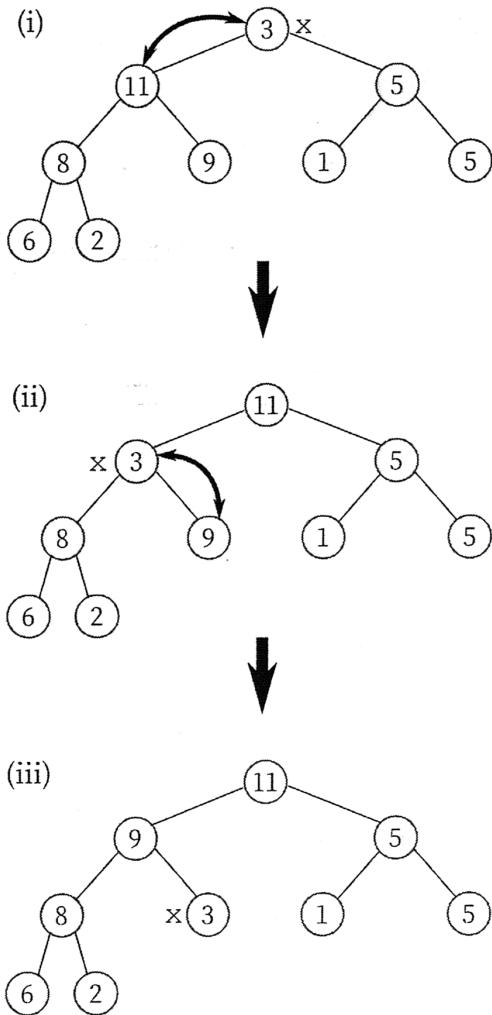


図 3.3: ヒープ木を構成する例

```

void swap(int a[], int i, int j) {
    int k;
    k = a[i]; a[i] = a[j]; a[j] = k;
}

void func1(int a[], int i, int j) {
    int k;
    while ((k = [ア ]) <= j) {
        if (k < j && [イ ]) {
            k++;
        }
        if (a[i] >= a[k]) {
            break;
        }
        swap(a, i, k);
        [ウ];
    }
}

void func2(int a[], int m) {
    int i;
    for ([エ] ; [オ] ; i--) {
        func1(a, i, [カ]);
    }
}

void func3(int a[], int m) {
    [キ];
    while (m > 1) {
        [ク];
        func1(a, [ケ], m - 2);
        m--;
    }
}

```

図 3.2: ヒープソートの関数

4.

図 4.1 に示すフィードバック制御系について、以下の問い合わせに答えよ。

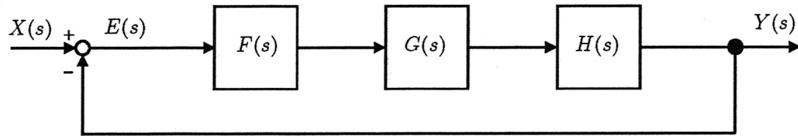


図 4.1

ただし、

$$F(s) = K_P \left(1 + \frac{K_I}{s} + K_D s \right), \quad K_P, K_I, K_D \geq 0,$$

$$G(s) = \frac{1}{s^2 + 2s + 4}, \quad H(s) = \frac{1}{s}$$

とする。

- 1) 以下の (ア), (イ), (ウ) をそれぞれ求めよ。分母, 分子ともに展開した上で s の降べきの順に整理して書くこと。
 - (ア) 開ループ伝達関数 $P(s)$
 - (イ) 閉ループ伝達関数 $Q(s)$
 - (ウ) $X(s)$ を入力とし $E(s)$ を出力とする偏差伝達関数 $R(s)$
- 2) $H(s)$ のボード線図 (Bode plot) を描け。横軸は角周波数 ω の対数とし、縦軸はゲインをデシベル (dB), 位相を度 ($^\circ$) で表すこと。また、適切な箇所に適切な数値を記入すること。
- 3) $K_I = K_D = 0$ のとき、閉ループ伝達関数 $Q(s)$ について、次のラウス表 (Routh table) の空欄 (エ), (オ) をそれぞれ埋めよ。

s^3	1	4	
s^2	2	K_P	
s^1	(エ)	0	
s^0	(オ)		

- 4) $K_I = K_D = 0$ のとき、系が安定となる K_P の範囲を求めよ。
- 5) $K_I = K_D = 0$ で系が安定であるとき、単位ランプ入力に対する定常偏差を求めよ。
- 6) $K_D = 0$ で系が安定であるとき、単位ランプ入力に対する定常偏差を求めよ。
- 7) P 制御に対する PI 制御の利点、および PI 制御に対する PID 制御の利点について簡単に説明せよ。

5. 以下の間に答えよ.

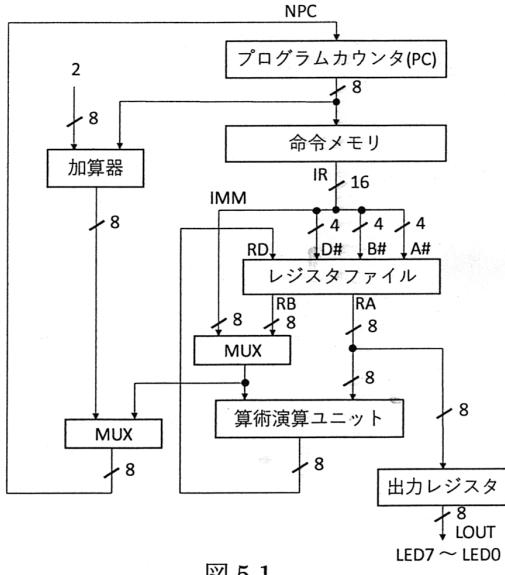


図 5.1

命令	ニーモニック形式	レジスタ転送式
加算	ADD RD, RB, RA	$R[D\#] \leftarrow R[B\#] + R[A\#]$
減算	SUB RD, RB, RA	$R[D\#] \leftarrow R[B\#] - R[A\#]$
結果出力	OUT RA	出力レジスタ $\leftarrow R[A\#]$
ジャンプ	J IMM	$PC \leftarrow IMM$
分岐	B RB, RA	$if (R[A\#] \neq 0) PC \leftarrow R[B\#]$
即値設定	SET RD, IMM	$R[D\#] \leftarrow IMM$

図 5.2

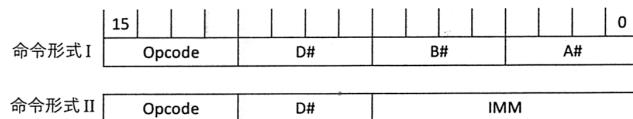


図 5.3

- 1) 図 5.1 に示す 8 ビットプロセッサを用いて、8 個の LED(発光ダイオード)を制御する。このプロセッサの主な構成要素は、16 ビット幅の命令 128 個を格納できる命令メモリ、8 ビット幅のレジスタ 16 個(R0～R15)を含むレジスタファイル、算術演算ユニットである。プログラムカウンタ、出力レジスタ、レジスタファイルに含まれる全てのレジスタはクロックの立ち上がりのタイミングで更新され、それらの初期値を 0 とする。なお、配線遅延やゲート遅延は無視できるものとする。

図 5.2 に命令仕様を示す。ニーモニック形式では、RD, RB, RA はそれぞれデスティネーションレジスタ、ソースレジスタ B, ソースレジスタ A を表し、R0～R15 の任意のレジスタを指定できる。レジスタ転送式では、D#はデスティネーションレジスタの番号、B#はソースレジスタ B の番号、A#はソースレジスタ A の番号を表す。レジスタ転送式における $R[n]$ はレジスタ番号 n で指定されるレジスタが保持するデータを表す。左矢印記号 \leftarrow はデータ転送である。IMM は命令に含まれる 8 ビット即値である。即値とレジスタファイルに格納される値は符号無し整数とする。

結果出力命令は RA の値を出力レジスタに書き込み、書き込まれた値の各ビットが LED7～LED0 として出力される。ジャンプ命令は即値 IMM をプログラムカウンタに転送する。分岐命令は、RA の値が 0 でない場合に RB の値をプログラムカウンタに転送し、そうでない場合にプログラムカウンタの値に 2 を加算する。ジャンプと分岐を除く命令では、プログラムカウンタの値に 2 を加算する。各命令はすべて 1 クロックで実行される。

図 5.3 に命令形式を示す。加算、減算、結果出力、分岐の各命令は命令形式 I、ジャンプ、即値設定は命令形式 II を採用する。

(次ページにつづく)

(前ページのつづき)

	LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
Clock 3	○	○	○	●	○	○	○	●
Clock 6	○	○	●	○	○	○	●	○
Clock 9	○	●	○	○	○	●	○	○
Clock 12	●	○	○	○	●	○	○	○
Clock 15	○	○	○	●	○	○	○	●
Clock 18	○	○	●	○	○	○	●	○
Clock 21	○	●	○	○	○	●	○	○
Clock 24	●	○	○	○	●	○	○	○

図 5.4

00:	SET R10, 11
02:	OUT R10
04:	ADD R10, R10, R10
06:	J 08
08:	OUT R10
0a:	(ア)
0c:	J 0e
0e:	OUT R10
10:	(イ)
12:	J 14
14:	OUT R10
16:	J 00

図 5.5

	A#	B#	D#	RA	RB	RD	NPC	LOUT
Clock 1	—	—	a	—	—	11	02	00
Clock 2	a	—	—	11	—	—	04	00
Clock 3	a	a	a	11	11	22	06	11

図 5.6

- (a) 図 5.4 のパターンで LED の点滅を繰り返すプログラムを考える。出力 LED0～LED7 が 8 個の LED を制御する。出力レジスタの LSB が LED0 に対応する。

各出力のビットが 1 の時に LED が点灯し、これを記号●を用いて表す。0 の時に LED が消灯し、これを記号○を用いて表す。図 5.5 は、図 5.4 の点滅を実現するプログラムである。行頭の数字はアドレスである。即値は 16 進数で記述している。空欄(ア), (イ)を埋めよ。

- (b) 図 5.5 のプログラムを実行する時、4 クロック目(アドレス 06 のジャンプ命令が実行される)から 12 クロック目までの 9 命令の、図 5.1 における A#, B#, D#, RA, RB, RD, NPC, LOUT の値を 16 進数で示せ。NPC はプログラムカウンタの入力、LOUT は出力レジスタの出力である。図 5.6 に 1 から 3 クロック目の値を示す。同様の形式にて回答すること。NPC と LOUT を除き、命令実行において意味を持たない箇所には記号 — を記入すること。

(次ページにつづく)

(前ページのつづき)

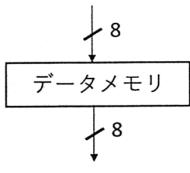


図 5.7

命令	ニーモニック形式	レジスタ転送式
加算	ADD RD, RB, RA	$R[D\#] \leftarrow R[B\#] + R[A\#]$
減算	SUB RD, RB, RA	$R[D\#] \leftarrow R[B\#] - R[A\#]$
結果出力	OUT RA	出力レジスタ $\leftarrow R[A\#]$
ジャンプ	J IMM	$PC \leftarrow IMM$
分岐	B RB, RA	if ($R[A\#] \neq 0$) $PC \leftarrow R[B\#]$
即値設定	SET RD, IMM	$R[D\#] \leftarrow IMM$
ロード	LD RD, RB, RA	$R[D\#] \leftarrow M[R[B\#]] + R[A\#]$

図 5.8

命令メモリ	データメモリ
00: SET R1, 01	20: 01
02: SET R2, 02	21: 10
04: SET R13, 20	22: 02
06: SET R14, 0c	23: 28
08: SET R10, 08	24: 04
0a: SET R11, 01	25: 44
0c: LD R8, R13, R11	26: 08
0e: OUT R8	27: 44
10: ADD R11, R11, R2	28: 10
12: SUB R10, R10, R1	29: 7c
14: B R14, R10	2a: 20
16: J 08	2b: 44
	2c: 40
	2d: 44
	2e: 80
	2f: 00

図 5.9

- 2) 図 5.7 に示す 8 ビットのデータ 256 個を格納できるデータメモリからレジスタにデータをロードするロード命令を追加する。図 5.8 は、図 5.2 にロード命令を加えた命令仕様である。レジスタ転送式における $M[a]$ はデータメモリのアドレス a に格納されているデータを表す。このロード命令も 1 クロックで実行される。

- (a) 図 5.1 を修正して、この仕様のプロセッサの全体構成を図示せよ。
 (b) このプロセッサで図 5.9 のプログラムを実行する。9, 14, 19, 24, 72, 77, 82, 87 クロック目における LED の点灯の様子を図 5.4 と同様の形式で示せ。

5 (補足説明)

図 5.1 では各構成要素を制御するための信号線は省略されている。

2) (a) 全体構成を図示する時は、図 5.1 と同様に制御のための信号線を省略してよい。