大阪大学大学院情報科学研究科
平成 30 年度　博士前期課程　入試問題
A 情報工学　解答・解説

楠本研究室：田中紘都，土居真之，松本淳之介，林純一

2016 年 XX 月 XX 日

1　アルゴリズムとプログラミング

■■■ 解答 ■■■

(1)

(1-1) 図 1 参照.

(1-2) 引数 x に対して，x が最上位の上司の構成員番号なら x を，そうでなければ x の直属の上司の構成員番号を返す.

(1-3) A. `find(y)`

B. `n==m` もしくは `m==n`

(2)

(2-1) $O(h)$

ノードを辿る回数は高々 h の定数倍であるため.

(2-2)

(2-2-1) `return p[x] = find(p[x]);`

(2-2-2) $O(1)$

関数 same を十分大きな回数実行すると，各々の木が深さ 1 とみなすことができるため.

■■■ 解説 ■■■

(1)

(1-1) 図 1 のような解答になる．実際の試験では 5 の存在を忘れている人が多数いた．高さ 0 でも木なので忘れないように.

(1-2) 問題文とプログラムを見比べながらそれっぽく記述すればよい.

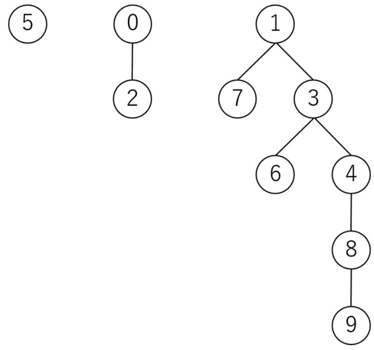

図 1　(1-1) の木

(1-3) `same(x,y)` という関数が問題文やプログラム中のコメントより `x` と `y` が同じ組織に所属するかを判定し出力する関数であることがわかる．そして 13 行目では `n=find(x)` が実行されていることから `n` は `x` の最上位の上司の構成員番号であることがわかる．こういった点から `m=find(y)` とし，最上位の上司が同じであれば，すなわち `m==n` であれば，`x` と `y` は同じ組織に所属すると判定できると推測していく.

また B に関しては勿論 `n-m==0` 等も正解になると思われる.

(2)

(2-1) 解答の通り，関数 `same` 内でループ処理を行っているのは関数 `find` のみであり，関数 `find` はノードを辿る操作として再起を行う．この再起の回数は

根から引数である構成員番号までの高さであり，その木の高さの平均は h であるため，関数 `same` の平均時間計算量は $O(h)$ である．

(2-2-1)

問題文に「配列 p を更新することで」とあることから代入文が用いられると推測できる．また，(2-1) より木の高さが小さければ小さいほど時間計算量はよくなるため，木の高さを小さくするような代入を行えばよい．以上のことから解答のような文を入れればよい．代入文であることに気を取られて `return` や `;` を忘れないように．

(2-2-2) 十分な回数実行すると，(2-2-1) の動作よりほぼすべての構成員番号に対して高さは 1 とみなすことができるため，関数 `find` は再起処理を行うことなく $O(1)$ で最上位の上司の構成員番号を返すようになる．したがって関数 `same` も $O(1)$ で実行することができる．

■■■ 所感 ■■■

今年の問題は Union-Find と呼ばれるデータ構造の問題でした．授業ではおそらく扱っていないデータ構造なので，例年よく出るソートや，そろそろ出そうな文字列探索などを勉強していた多くの受験生はかなり焦りました．しかし問題自体は例年に比べとても簡単で，完答した人も多かったようです．実際にこの大問の平均点は例年より高かったとか...

また従来の問題との変更点として，データがテキストで与えられ，fscanf で読み込むようになっていました．

2　計算機システムとシステムプログラム

■■■ 解答 ■■■

(1)

(1-1) 時間的局所性
空間的局所性

(1-2) (a) 頻繁にアクセスするアドレスに対して高速にアクセスが可能になる．
(b) メインメモリの容量以上のサイズのプロセスも割り付けが可能となる．

(1-3)

(1-3-1) この設問自信なし．
(a) 8 (b) 4 (c) 32 (d) 2 (e) 2 (f) 5

(1-3-2)

	0	1	2	3	4	5	6	7
(a)	0	0	0	0	0	1	2	3
(b)	0	0	1	1	2	0	2	3
(c)	0	0	0	0	0	1	2	31
(d)	25%							

(2)

(2-1) (ア) F (イ) B (ウ) E
(エ) G (オ) A (カ) I

(2-2)

(2-2-1) 表 2 参照．
(2-2-2) 図 2, 図 3 参照．
(2-2-3) 1 対 1 に近づく．

■■■ 解説 ■■■

(1)

(1-1) 「時間的局所性」と「空間的局所性」は頻出ワードであるため確実に抑えておきたい．

(1-2) 省略．キャッシュについて自身のない人は H25 や H21，仮想記憶に自信のない人は H27 や H24 や H18 など定期的に出題されているためその年度の問題も解いて勉強しましょう．

(1-3)

(1-3-1)

(a) 問題文「アドレスは 1[バイト]([byte]) 毎に付与される．」より 1byte．単位は bit なので 8bit．

(b)(キャッシュメモリ内のブロック数)=(キャッシュのサイズ)/(ブロックのサイズ) であり，キャッシュのサイズは 8[バイト]，ブロックのサイズは 2[バイト] なので，$8 \div 2 = 4$.

(c) (ページテーブルのエントリ数)=(ページテーブルのサイズ)/(1 ページのサイズ) であり，ページテーブルのサイズは仮想記憶のサイズと同じであるため，$256 \div 8 = 32$.

(d) (主記憶内のページ数)=(主記憶のサイズ)/(1 ページのサイズ) であるため，$32 \div 8 = 4 = 2^2$. よって 2bit.

(e) (b) よりブロック数は 2^2 なので，インデックスに 2bit 必要．

(f) (a) よりアドレスは 8bit で指定される．(b) よりブロック数は 2^2 なので，インデックスに 2bit 必要．問題文よりブロックサイズは 2^1 バイトなのでオフセットに 1bit 必要．よってタグ長は $8 - 2 - 1 = 5$bit.

(1-3-2) (e),(f) よりタグ，インデックス，オフセットの長さはそれぞれ 5bit，2bit，1bit であるから，参照される様子は表 1 のようになる．表 1 をもとに答えればよい．

(2)

(2-1) 省略．覚えていないものがあれば確認しておこう．

(2-2)

(2-2-1)

解答は表 2 のようになる．各ファイル割付の手法に関する解説はここでは省略する．各自確認してください．

表 1　(1-3-2)

参照論理アドレス				hit/miss	アクセスされるブロックのタグ (2 進数)			
10 進数	tag	index	offset		0(00)	1(01)	2(10)	3(11)
0	00000	00	0	MISS	0(00000)			
1	00000	00	1	HIT	”			
2	00000	01	0	MISS	”	0(00000)		
3	00000	01	1	HIT	”	”		
4	00000	10	0	MISS	”	”	0(00000)	
8	00001	00	0	MISS	1(00001)	”	”	
21	00010	10	1	MISS	”	”	2(00010)	
255	11111	11	1	MISS	”	”	”	31(11111)

表 2　(2-2-1)

	(i)	(ii)	(iii)
(a)	$ceil(\frac{n}{b})$	$ceil(\frac{n}{b-a})$	2
(b)	$ceil(\frac{n}{b})$	$ceil(\frac{n}{b-a})$	$ceil(\frac{n}{b})+1$
(c)	$b \cdot ceil(\frac{s}{b})$	$b \cdot ceil(\frac{s}{b-a})$	$b \cdot (ceil(\frac{s}{b})+1)$

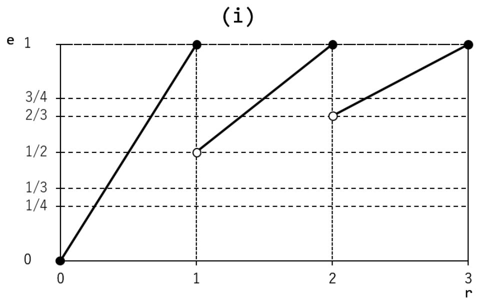

図 2　(2-2-2) の (i)

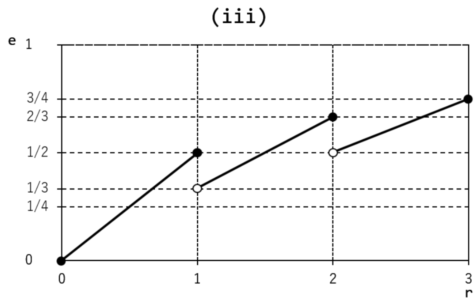

図 3　(2-2-2) の (iii)

(2-2-2)

問題文中の $e=\frac{s}{p}$ に (2-2-1) の結果と $r=\frac{s}{b}$ を代入していくと，(i) は $e=\frac{r}{ceil(r)}$，(ii) は $e=\frac{r}{ceil(r+1)}$ となる．よって解答としては図 2 と図 3 のようになる．ちなみに実際の解答用紙にも，枠線や点線が書かれていた．

(2-2-3)

(2-2-2) のグラフをよく見ると e が r が大きくなるにつれてほぼ 1 になることがわかるので，それを記述すればよい．

所感

今回はキャッシュとページングの仮想記憶からの出題と，過去出題されたことのないファイルシステムからの出題で，計算量が多く時間内に解こうとするには非常に難易度が高い大問でした．実際試験後に私をはじめ多くの受験生が口をそろえて大問 2 が難しかったと嘆いていました．ただ (2) は特に前提知識さえあれば，時間次第では解ける問題であったため，こういった問題に柔軟に対応できるようにしておくといいでしょう．

ちなみにこの大問 2 ですが，出題の傾向が数年前

の過去問から変わりつつあります．というのも，例年この大問を作成していたと思われる今井先生が昨年度で定年退職され新しい先生 (橋本先生かな) になったためです．今井先生の場合，計算機アーキテクチャの期末試験と問題が一致したことがあったりしますが，今後はそういったことは期待できないということを覚えておくといいかもしれません．

3　離散構造

■■■ 解答 ■■■

(1)

(1-1) $\forall X\ append([\,], X, X)$

(1-2) $\forall A\ \ \forall X\ \ \forall Y\ \ \forall Z\ \ (append(X, Y, Z) \rightarrow append([A|X], Y, [A|Z]))$

(1-3) $reverse([\,], [\,])$

(1-4) α: $[A|X]$

(1-5)

(1-5-1) β: $\displaystyle\bigwedge_{1\leq i\leq m} (\neg P_{i1} \vee \cdots \vee \neg P_{in_i} \vee Q_i)$

γ: $(\neg R_1 \vee \cdots \vee \neg R_k)$

(1-5-2) 分かりませんでした...

(2)

(2-1)

(2-1-1) $f_n = 2^n - 1$

(2-1-2)
- n, i の一方が偶数，もう一方が奇数の場合：棒 2
- n, i ともに偶数もしくは，ともに奇数の場合：棒 3

(2-2)（あまり自信なし...）

(2-2-1) $g_n = 2^{d+1} + 2^{n-d} - 3$

(2-2-2) $d = n - 1$

■■■ 解説 ■■■

(1)

(1-5-1)

$$式 (1) = (\bigwedge_{1\leq i\leq m} (\forall x_1 \cdots \forall x_h((P_{i1} \wedge \cdots \wedge P_{in_i}) \rightarrow Q_i))) \rightarrow \exists x_1 \cdots \exists x$$

$$= \neg(\bigwedge_{1\leq i\leq m} (\forall x_1 \cdots \forall x_h(\neg(P_{i1} \wedge \cdots \wedge P_{in_i}) \vee Q_i))) \vee (\exists x_1 \cdots$$

よってこの否定は

$$(\bigwedge_{1\leq i\leq m} (\forall x_1 \cdots \forall x_h(\neg(P_{i1} \wedge \cdots \wedge P_{in_i}) \vee Q_i))) \wedge \neg(\exists x_1 \cdots \exists x_h(P$$

$$= (\bigwedge_{1\leq i\leq m} (\forall x_1 \cdots \forall x_h(\neg P_{i1} \vee \cdots \vee \neg P_{in_i} \vee Q_i))) \wedge (\forall x_1 \cdots \forall x_h(\neg R$$

$$= \forall x_1 \cdots \forall x_h((\bigwedge_{1\leq i\leq m} (\neg P_{i1} \vee \cdots \vee \neg P_{in_i} \vee Q_i)) \wedge (\neg R_1 \vee \cdots \vee \neg R$$

(2)

(2-1)

(2-1-1)

n 枚の円盤を移動させるためには，まず 1 枚目から $n-1$ 枚目までの円盤を異動させる必要がある．つまり

- 円盤 1〜$n-1$ を棒 2 に移動
- 円盤 n を棒 3 に移動
- 円盤 1〜$n-1$ を棒 3 に移動

という過程となる．

ここで，$n-1$ 枚目までの円盤の移動回数は f_{n-1} と表せ，n 枚目の円盤の移動回数は 1 回である．よって，f_n についての漸化式は

$$\begin{aligned} f_n &= f_{n-1} + 1 + f_{n-1} \\ &= 2f_{n-1} + 1 \\ f_n + 1 &= 2(f_{n-1} + 1) \end{aligned}$$

以上より，数列 $f_n + 1$ は初項 2，公比 2 の等比数列より

$$\begin{aligned} f_n + 1 &= 2^n \\ f_n &= 2^n - 1 \end{aligned}$$

(2-2-1)

条件を満たす円盤の移動手順は

- 円盤 1〜d までを棒 3 に移動
- 円盤 $d+1$〜$n-1$ を棒 2 に移動
- 円盤 n を棒 4 に移動
- 円盤 $d+1$〜$n-1$ を棒 4 に移動
- 円盤 1〜d までを棒 4 に移動

である．

ここで， 円盤 1〜d は棒 2 に挿されてはいけないことから，最小の移動回数は $m=3$ における円盤 d 枚の場合の移動回数と等しくなる．つまり (2-1-1) より

$$2^d - 1[\text{回}]$$

同様に円盤 $d+1$〜$n-1$ は棒 3 に挿されてはいけないことから，最小の移動回数は $m=3$ における円盤 $n-d-1$ 枚（$(n-1)-(d+1)+1 = n-d-1$ より）の場合の移動回数と等しくなる．つまり (2-1-1) より

$$2^{n-d-1} - 1[\text{回}]$$

以上より g_n は

$$\begin{aligned} g_n &= (2^d - 1) + (2^{n-d-1} - 1) + 1 + (2^{n-d-1} - 1) + (2^d - 1) \\ &= 2(2^d - 1) + 2(2^{n-d-1} - 1) + 1 \\ &= 2^{d+1} + 2^{n-d} - 3 \end{aligned}$$

(2-2-2)

(2-1-1) より $f_n = 2^n - 1$，(2-2-1) より $g_n = 2^{d+1} + 2^{n-d} - 3$．以上より，

$$\begin{aligned} f_n &= g_n \\ 2^n - 1 &= 2^{d+1} + 2^{n-d} - 3 \\ 2^n + 2 &= 2^{d+1} + 2^{n-d} \end{aligned}$$

この等式が成立することから

$$n = d + 1$$

もしくは

$$n = n - d$$

が成り立つ．よって $d = 0, n-1$ となるが，条件より $d = 0$ は不適切．よって

$$d = n - 1$$

4　計算理論

■■■ 解答 ■■■

(1)

(1-1) (i) 7　(ii) 8　(iii) 5　(iv) 3

(1-2)

(1-2-1) 図4 を参照.

(1-2-2) (2, 2) / 1
(1, 1) / 2
(1, Z) / 1Z
(2, Z) / 2Z

(2)(2-1) (), ()(), (()), ((())), ()(()), (())(), ()()()

(2-2) 図4 を参照.

(2-3) 5

(2-4) 14

(2-5) (ア) ()　(イ)$AA \xrightarrow{*} vw$　(ウ)(y)
(エ)$(A) \xrightarrow{*} (y)$

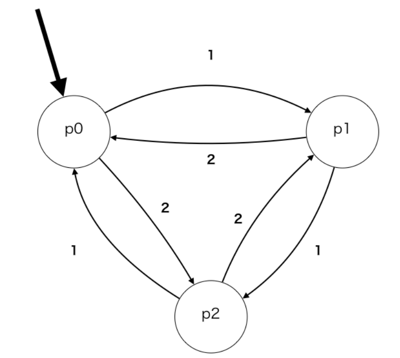

図 4　(1-2-1) の状態遷移図

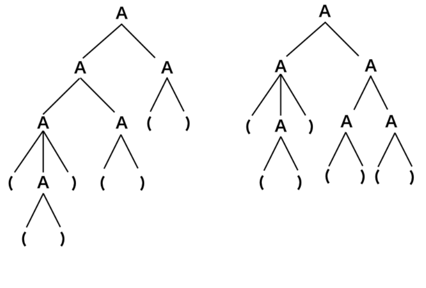

図 5　(2-2) の構文木

■■■ 解説 ■■■

(1)

(1-1)

最短で受理状態に行くにはどのような遷移が必要かを考え，その最短のルートと正則表現の最短の文字が一致するかで選択肢を一気に減らしてしまうと考えやすい問題である．選択肢を減らした後は状態遷移図で受理できない語を含む正則表現を除外し，残った一つが答えとなる (ただし念のため答えの正則表現の語をいくつか用意し，正しく受理できるか確認すること).

(ア)

最短で受理状態に遷移するものは aa である．選択肢の正則表現の中で最短の文字が aa であるものは 7 のみである．

(イ)

最短で受理状態に遷移するものは (a + b) である．選択肢の正則表現の中で最短の文字が (a + b) であるものは 1, 8 である．1 であると aa も受理できなければならないが，(イ) の状態遷移図はこれを受理することはできない．よって 8 である．

(ウ)

最短で受理状態に遷移するものは ab である．選択肢の正則表現の中で最短の文字が ab であるものは 5 のみである．

(エ)

最短で受理状態に遷移するものは ε である．選択肢の生息表現の中で最短の文字が ε であるものは 3, 4 である．4 であると ba も受理できなければならないが，(エ) の状態遷移図はこれを受理することはできない．よって 3 である．

(1-2)

(1-2-1)

p_0 が受理状態であるので，p_0 を 3 で割った余りが 0 の状態とし．p_1，p_2 をそれぞれ 3 で割った余りが 1, 2 の状態とすれば簡単に遷移図を書くことはできる．

(1-2-2)

スタックのトップで現在読み込んだ入力の余りを表現する．つまり，スタックのトップが Z であれ

ば受理状態，1 であれば 3 で割った余りが 1 の状態，2 であれば 3 で割った余りが 2 の状態である．入力アルファベットが 2 通り，状態が 3 通りなので，全部で 6 通りの入力と状態の組み合わせが存在し，問題用紙にすでに 2 つは記入してあるので，残りの 4 組を答えればよい．あらかじめ入力と状態の組み合わせが何通り存在するか計算しておけば，記入漏れを防ぐことができる．

(2) 略

■■■ 所感 ■■■

比較的易しい問題が多く，短時間で点数を取りにいく大問であった．(2-3) や (2-4) のように自分の答えが網羅できているか不安になるかもしれないが，たかが 10 点の小問であるので，何度も確認したりせず一度回答したら潔く次に進むべきである．この単元に限ったことではないが配点が低く次の問題に影響しない問題に必要以上に悩むことは無駄であり，そのような時間があるのであれば，配点の高い問題を考え確実に正解することの方が大事であると考える．

6 電子回路と論理設計

■■■ 解答 ■■■

(1)

(1-1) $A = (0, 1, 1)$, $B = (0, 1, 0)$ のとき，
$A' = (0, 0, 1, 1)$, $B' = (0, 0, 1, 0)$,
$T = (0, 0, 0, 1)$, $F = (0, 0, 1)$.
$A = (1, 0, 1)$, $B = (0, 1, 0)$ のとき，
$A' = (1, 1, 0, 1)$, $B' = (0, 0, 1, 0)$,
$T = (1, 0, 1, 1)$, $F = (1, 0, 1)$.

(1-2) $a_3 = a_2$, $b_3 = b_2$

(1-3) $x_0 = a_0$, $x_1 = a_1$, $x_2 = a_2$, $x_3 = a_3$,
$y_0 = \overline{b_0}$, $y_1 = \overline{b_1}$, $y_2 = \overline{b_2}$, $y_3 = \overline{b_3}$,
$z_0 = 1$, $z_1 = c_0$, $z_2 = c_1$, $z_3 = c_2$

(1-4) $f_2 = t_2\,\overline{t_1}\,\overline{t_0} + \overline{t_3}\,t_2 + t_3\,\overline{t_2}$,
$f_1 = t_3\,\overline{t_1}\,t_0 + \overline{t_3}\,t_1 + t_1\,\overline{t_0}$,
$f_0 = t_0$

(2)

(2-1) A. pMOS　B. pMOS
C. nMOS　D. nMOS

(2-2) $2C_{\mathrm{L}}R_{\mathrm{MOS}}\ln 2$

■■■ 解説 ■■■

(1)

(1-1)

A', B' は元の数 A, B とそれぞれ同値であるから，A, B を 10 進数にし，それぞれを 4 bit の 2 の補数表現にすれば良い．(もちろん (1-2) の結果がすぐに思い付けばそれでも良い．)

T は $A' - B'$ を計算すれば良い．

F は T の絶対値を 3 bit の符号なし整数にしたものであるから，T が非負なら T の下位 3 bit，T が負なら $-T$ の下位 3 bit になる．

(1-2)

(1-1) の A' が求められれば気が付くと思うが，A' の最上位ビット (MSB) は，A が非負なら 0，負なら 1 となる (B も同様)．よって，A', B' の MSB a_3, b_3 は A, B の MSB a_2, b_2 とそれぞれ等しい．その他の部分の各ビットは A, B とそれぞれ同じで

ある．

(1-3)

$T = A' - B' = A' + (-B')$ であるから，A' はそのまま，B' は符号反転して足し合わせれば良い．2 の補数表現された整数を符号反転するには，各ビットを反転したものに 1 加えれば良い．この操作を全加算器を用いて実現する．

したがって，各全加算器 FA_i の入力を，A' の a_i，B' の否定をとった $\overline{b_i}$，前段の全加算器 FA_{i-1} $(i > 0)$ の桁上げ出力 c_{i-1} とし，最下位ビットの和を計算する全加算器 FA_0 の桁上げ入力を 1 とすれば良い．

なお，問題文において全加算器 FA_i の入力 x_i, y_i, z_i の役割 (特に桁上げ入力) が指定されていないが，適当に決めて良いと思われる．解答例では z_i を桁上げ入力とした．

(1-4)

真理値表は表 3 のようになる．$T = (1, 0, 0, 0)$ は起こり得ないので，このときの F はすべて don't care (表 3 には d と表記) とする．

真理値表を基に f_j $(j = 0, 1, 2)$ のカルノー図 (図 6) をかき，最簡積和形を求めれば良い．

(2)

(2-1) 省略．

(2-2)

図 3 (問題用紙参照) の MOSFET のうち， A, B を抵抗 R_{MOS} に置き換え， C, D を開放した (取り除いた)，CR 直列回路の過渡応答の問題である．

入力 x_0, x_1 が V_{DD} から 0 に変化した瞬間の時刻を $t = 0$ として，時刻 t における C_{L} の瞬時電荷を $q(t)$，C_{L} にかかる電圧を $v(t)$，回路に流れる電流を $i(t)$ とする．このとき，$q(t) = C_{\mathrm{L}} v(t)$, $i(t) = \dfrac{dq(t)}{dt}$，およびキルヒホッフの電圧則から，$q(t)$ に関する微分方程式

$$2R_{\mathrm{MOS}} \frac{dq(t)}{dt} + \frac{1}{C_{\mathrm{L}}} q(t) = V_{\mathrm{DD}}$$

が得られる．初期条件 $q(0) = 0$ のもとで上記微分方程式を解き，$v(t)$ を求めると

$$v(t) = V_{\mathrm{DD}} \left(1 - \exp\left(-\frac{t}{2C_{\mathrm{L}} R_{\mathrm{MOS}}}\right)\right)$$

となる．最後に，$v(t) = 0.5V_{\mathrm{DD}}$ を t について解けば，解答が得られる．

所感

論理設計の問題は組み合わせ論理回路の設計でした．状態遷移などがない分簡単だったと思います．最簡積和形を求める問題は毎年出ているので，焦ってカルノー図の書き方・囲み方など間違えないよう練習しておくと良いと思います．

電子回路の問題は MOSFET を用いた NOR 回路についてでした．過去には，H24 に CMOS 型 NAND 回路について今回の (2-2) と同様の問題が出題されています．電子回路の問題は例年論理設計より配点が少なく設定されており，そもそも出題されない年もあるようですが，MOSFET の構造や原理 (H27 に出題)，それを用いた CMOS 型論理ゲート (H28, H27, H25 に出題) を抑えておけば解ける程度の問題だと思うので，勉強しておきましょう．また，今回は微分方程式を解く問題が出題されました．電子回路の範囲では，(論理回路を関連させる限り) 微分方程式は 1 階の限られたパターンしか出てこないので，余裕があれば解けるようにしておきましょう (CR 直並列回路ぐらいだと思うので覚えてしまうのも手だと思います)．

どちらもそれほど難しい問題ではなかったと思います．筆者の個人的な意見ですが，他の選択問題よりも得点しやすいと思います．基本を抑えてしっかり得点できるようにしておきましょう．

表 3　(1-4) 真理値表

t_3	t_2	t_1	t_0	f_2	f_1	f_0	t_3	t_2	t_1	t_0	f_2	f_1	f_0
0	0	0	0	0	0	0	1	0	0	0	d	d	d
0	0	0	1	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	1	0	1	0	1	0	1	1	0
0	0	1	1	0	1	1	1	0	1	1	1	0	1
0	1	0	0	1	0	0	1	1	0	0	1	0	0
0	1	0	1	1	0	1	1	1	0	1	0	1	1
0	1	1	0	1	1	0	1	1	1	0	0	1	0
0	1	1	1	1	1	1	1	1	1	1	0	0	1

f2

t3 t2 \ t1 t0	00	01	11	10
00				
01	1	1	1	1
11	1			
10	d	1	1	1

f1

t3 t2 \ t1 t0	00	01	11	10
00			1	1
01			1	1
11		1		1
10	d	1		1

f0

t3 t2 \ t1 t0	00	01	11	10
00		1	1	
01		1	1	
11		1	1	
10	d	1	1	

図 6　(1-4) カルノー図