

情報工学

予想問題 第7回

(ハードウェア設計重視年)

必須問題：問題 1・2 選択問題：問題 4・6・7 より 2 問選択

試験時間：3 時間 配点：500 点満点

2006～2026 年度の過去問分析に基づく自作予想問題

作成：Claude (Anthropic)

1 【必須問題】 アルゴリズムとプログラミング

配点：(1-1) 15 点, (1-2) 20 点, (2-1) 20 点, (2-2) 20 点, (3-1) 20 点, (3-2) 15 点, (3-3) 15 点

次ページの図 1 は ANSI C で記述されたプログラム (program) であり, 配列 (array) を用いた 2 分ヒープ (binary heap) を実装している。添字 i の要素の親は $\lfloor (i-1)/2 \rfloor$, 左の子は $2i+1$, 右の子は $2i+2$ である。大域変数 `count` は 21 行目が実行された回数を数えるカウンタである。図 1 のプログラムに関する以下の各問に答えよ。

(1) ヒープの構築について以下の各小問に答えよ。

(1-1) 関数 `funcA` が満たそうとしているヒープの性質を, 親と子関係として述べよ。また, プログラムが正しく動作するように空欄 (a) を埋めよ。

(1-2) 33 行目の `build` が完了した時点での配列 `A` の内容を示せ。また, その状態を 2 分木として図示せよ。

(2) プログラムの実行について以下の各小問に答えよ。

(2-1) 図 1 のプログラムが開始してから終了するまでに出力される内容をすべて記述せよ。

(2-2) 関数 `extract` を 3 回呼び出す過程において, 1 回目と 2 回目の呼び出し直後の配列 `A` の内容 (有効な範囲のみ) をそれぞれ示せ。

(3) 計算量と応用について以下の各小問に答えよ。

(3-1) 要素数 n のヒープに対する関数 `down` の最悪時間計算量 (worst-case time complexity) をオーダ表記で示し, その根拠を木の高さに基づいて 2, 3 行で説明せよ。

(3-2) 関数 `build` は, $i = n/2 - 1$ から 0 まで降順に `down` を呼び出す。この方法の最悪時間計算量は $O(n)$ であることが知られている。一方, 空のヒープに要素を 1 個ずつ挿入して構築する方法の最悪時間計算量は $O(n \log n)$ である。両者に差が生じる理由を, 木の各深さにある節点の個数と, そこから `down` が降下しうる段数の関係に基づいて 2, 3 行で説明せよ。

(3-3) このヒープを用いて, n 個の要素を降順に整列する方法を述べよ。また, その方法の時間計算量をオーダ表記で示せ。

```

1 #include <stdio.h>
2 #define N 7
3
4 int A[N];
5 int n;
6 int count = 0;
7
8 void swap(int *a, int *b) { int t = *a; *a = *b; *b = t; }
9
10 void down(int i) {
11     while (1) {
12         int l = 2 * i + 1;
13         int r = 2 * i + 2;
14         int m = i;
15         if (l < n && A[l] > A[m]) m = l;
16         if (r < n && A[r] > A[m]) m = r;
17         if (m == i) break;
18         swap(&A[i], &A[m]);
19         i = m;
20         count++;
21     }
22 }
23
24 void funcA(void) {
25     for (int i = n / 2 - 1; i >= 0; i--) down(i);
26 }
27
28 int extract(void) {
29     int top = A[0];
30     A[0] = A[n - 1];
31     n--;
32     down(0);
33     return top;
34 }
35
36 int main(void) {
37     int init[N] = {4, 10, 3, 5, 1, 8, 7};
38     for (int i = 0; i < N; i++) A[i] = init[i];
39     n = N;
40
41     funcA(); /* build */
42     for (int i = 0; i < n; i++) printf("%d ", A[i]);
43     printf("\n");
44
45     for (int i = 0; i < 3; i++) printf("%d ", extract());
46     printf("\n");
47     printf("%d\n", count);
48     return 0;
49 }

```

図1 プログラム

【2】 【必須問題】 計算機システムとシステムプログラム

配点：(1-1) 15 点, (1-2) 20 点, (1-3) 20 点, (1-4) 15 点, (2-1) 15 点, (2-2) 20 点, (2-3) 20 点

(1) キャッシュメモリ (cache memory) について以下の各小問に答えよ。対象とする計算機のアドレスは 32 ビット, バイトアドレッシング (byte addressing) である。キャッシュは容量 16 KiB, ブロックサイズ (block size) 64 バイト, 4 ウェイセットアソシアティブ方式 (4-way set-associative) とする。

(1-1) このキャッシュのセット数を求めよ。また, メモリアドレスをタグ (tag), インデックス (index), オフセット (offset) に分割したときの各ビット数を求めよ。

(1-2) キャッシュへのアクセス時間を 1 サイクル, キャッシュミス時のペナルティ (miss penalty) を 80 サイクルとする。ミス率 (miss rate) が 3% であるときの平均メモリアクセス時間 (AMAT: average memory access time) をサイクル単位で求めよ。

(1-3) 上記の構成に 2 次キャッシュ (L2 cache) を追加する。L2 キャッシュへのアクセス時間を 10 サイクル, L2 キャッシュのミス率 (1 次キャッシュでミスしたアクセスのうち L2 でもミスする割合) を 25%, 主記憶へのアクセスペナルティを 80 サイクルとする。このときの AMAT を求めよ。また, 小問 (1-2) の結果と比較して 2 次キャッシュの効果を 1, 2 行で述べよ。

(1-4) ブロックサイズを 64 バイトから 256 バイトに変更した場合を考える。キャッシュ容量とウェイ数を変えないとき, セット数とインデックスのビット数がどう変わるかを答えよ。また, ブロックサイズを大きくすることの長所と短所をそれぞれ一つずつ挙げよ。

(2) 並行プロセス (concurrent process) の排他制御 (mutual exclusion) について以下の各小問に答えよ。セマフォ変数 (semaphore) s に対する操作 $P(s)$, $V(s)$ を以下のように定義する。

- $P(s)$: $s > 0$ ならば s を 1 減らして続行する。 $s = 0$ ならば $s > 0$ になるまで待つ。
- $V(s)$: s を 1 増やす。

(2-1) 二つのプロセス P_1, P_2 が, 二つの資源 R_1, R_2 をそれぞれセマフォ s_1, s_2 (ともに初期値 1) で管理しながら使用する。以下のプログラムを並行実行したときにデッドロック (deadlock) が発生し得ることを, 具体的な実行順序を挙げて説明せよ。

プロセス P1	プロセス P2
$P(s_1);$	$P(s_2);$
$P(s_2);$	$P(s_1);$
/* R_1 と R_2 を使用 */	/* R_1 と R_2 を使用 */
$V(s_2);$	$V(s_1);$
$V(s_1);$	$V(s_2);$

(2-2) デッドロックが発生するための 4 つの必要条件をすべて挙げよ。さらに, 小問 (2-1) のプログラムについて, そのうちの一つの条件を崩すことでデッドロックを予防する具体的な修正を示し, その修正がどの条件を崩しているかを述べよ。

(2-3) 資源が 3 種類, プロセスが 3 個あるシステムにおいて, 各プロセスが必要とする資源の最大数と現在の割り当て数が表 1 のとおりであるとする。利用可能な資源の残数は $(A, B, C) = (3, 3, 2)$ である。

プロセス	最大要求数			現在の割り当て数		
	A	B	C	A	B	C
P_1	7	3	3	0	1	0
P_2	3	2	2	2	0	0
P_3	5	0	2	3	0	2

表1 各プロセスの資源要求と割り当て

各プロセスがあと必要とする資源数（最大要求数から現在の割り当て数を引いた値）をそれぞれ求めよ。
さらに、すべてのプロセスが完了できる実行順序が存在するかどうかを判定し、存在する場合はその順序を一つ示せ。このように安全な実行順序の有無を調べる手法の名称を答えよ。

4 【選択問題】 計算理論

配点：(1-1) 15 点, (1-2) 20 点, (1-3) 20 点, (2-1) 25 点, (2-2) 20 点, (3) 25 点

文脈自由文法 (context-free grammar) $G = (V, T, P, S)$ は, 変数 (variable) の有限集合 V , 終端記号 (terminal symbol) の有限集合 T , 生成規則 (production rule) の有限集合 P , および開始記号 (start symbol) S により定められる。 G が生成する言語を $L(G)$ と表す。

ある語 $w \in L(G)$ に対して, G における構文木 (parse tree) が 2 つ以上存在するとき, G は曖昧 (ambiguous) であるという。以下の各問に答えよ。

(1) 算術式を生成する文法 $G_1 = (\{E\}, \{+, \times, (,), \text{id}\}, P_1, E)$ を考える。 P_1 は以下のとおりである。

$$E \rightarrow E + E \mid E \times E \mid (E) \mid \text{id}$$

(1-1) 語 $\text{id} + \text{id} \times \text{id}$ に対する構文木を 2 つ示し, G_1 が曖昧であることを説明せよ。

(1-2) 小問 (1-1) の 2 つの構文木は, 演算子 $+$ と $\times$ のどちらを先に適用するかの違いに対応している。演算子の優先順位 (precedence) を $\times$ が $+$ より高いものとし, かつ同じ優先順位の演算子は左結合 (left-associative) とするとき, $L(G_1)$ と同じ言語を生成する曖昧でない文法 G_2 を示せ。

(1-3) 小問 (1-2) で構成した G_2 を用いて, 語 $\text{id} + \text{id} \times \text{id}$ の最左導出 (leftmost derivation) を示せ。また, 得られる構文木が一意であることを 1, 2 行で説明せよ。

(2) プッシュダウンオートマトン (pushdown automaton) は $(Q, \Sigma, \Gamma, \delta, q_0, Z, F)$ で表される。 Q は状態の有限集合, Σ は入力アルファベット, Γ はスタックアルファベット (stack alphabet), δ は遷移関数, q_0 は開始状態, $Z \in \Gamma$ はスタックの開始記号 (initial pushdown symbol), F は最終状態 (final state) の集合である。

遷移に付与するラベル $(r, s)/t$ ($r \in \Sigma \cup \{\varepsilon\}, s \in \Gamma, t \in \Gamma^*$) は, 入力から読み出す記号が r でスタックの先頭にある記号が s のとき, スタックからその s を取り去り t をスタックに押し込むことを意味する。 t が複数の記号の場合は右側の記号から順に押し込む。 r が ε の場合は入力から記号を読み出さずに遷移する。 t が ε の場合はスタックに記号を押し込まない。

(2-1) 言語 $L = \{a^i b^j c^k \mid i, j, k \geq 0, i = j \text{ または } j = k\}$ を, 最終状態による受理 (acceptance by final state) で認識するプッシュダウンオートマトンを構成したい。スタックアルファベットを $\Gamma = \{Z, X\}$ とし, 必要な遷移をすべて $(r, s)/t$ の形式で列挙せよ。また, どこで非決定性 (nondeterminism) を用いているかを 1, 2 行で述べよ。

(2-2) 言語 L は本質的に曖昧 (inherently ambiguous) であること, すなわち L を生成するどの文脈自由文法も曖昧であることが知られている。語 $a^n b^n c^n$ に対して 2 通りの解釈が生じる理由を, 小問 (2-1) で構成したオートマトンの動作に基づいて 2, 3 行で説明せよ。

(3) 文脈自由言語 (context-free language) の閉包性 (closure property) について考える。文脈自由言語は和集合 (union), 連接 (concatenation), クリーネ閉包 (Kleene closure) について閉じているが, 積集合 (intersection) および補集合 (complement) については閉じていない。

積集合について閉じていないことは, $\{a^n b^n c^m \mid n, m \geq 0\}$ と $\{a^m b^n c^n \mid n, m \geq 0\}$ の積集合が $\{a^n b^n c^n \mid n \geq 0\}$ となることから示される。この事実と, 文脈自由言語が和集合について閉じていることを用いて, 文脈自由言語が補集合について閉じていないことを背理法 (proof by contradiction) により証明せよ。(ヒント: ド・モルガンの法則 $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ を用いよ。)

6 【選択問題】電子回路と論理設計

配点：(1-1) 15 点, (1-2) 25 点, (1-3) 15 点, (2-1) 20 点, (2-2) 30 点, (3-1) 10 点, (3-2) 10 点

4 ビット符号なし 2 進数 (4-bit unsigned binary number) の被除数 N を除数 D で割り, 商 (quotient) Q と剰余 (remainder) A を求める専用ハードウェア (dedicated hardware) を設計する。用いるアルゴリズムは復元法 (restoring division) であり, その手順は以下のとおりである。

復元法による除算

5 ビットのレジスタ A (剰余) と 4 ビットのレジスタ Q を用いる。初期状態では $A = 00000_2$, $Q = N$ とする。以下を 4 回繰り返す。

1. レジスタ A と Q を連結した 9 ビットを 1 ビット左シフトする (Q の最上位ビットが A の最下位ビットに入り, Q の最下位ビットには後の手順で商のビットが入る)。
2. $A \leftarrow A - D$ を計算する。
3. 結果が 0 以上ならば Q の最下位ビットに 1 を入れる。結果が負ならば A を減算前の値に戻し (復元し), Q の最下位ビットに 0 を入れる。

4 回の繰り返しの後, Q に商が, A に剰余が得られる。

(1) アルゴリズムの動作について以下の各小問に答えよ。

(1-1) $N = 1011_2 (= 11)$, $D = 0011_2 (= 3)$ とする。初期状態におけるレジスタ A と Q の値を 2 進数で示せ。

(1-2) 小問 (1-1) の条件で, 4 回の繰り返しの各回について, 手順 1 のシフト直後の A の値, 手順 2 の減算結果の符号, および手順 3 の後の A と Q の値を表にまとめよ。最終的に得られる商と剰余を 10 進数で答え, $11 \div 3$ の結果と一致することを確認せよ。

(1-3) 手順 2 の減算結果が負であるかどうかは, どのようにして判定できるかを述べよ。また, そのとき A を復元する具体的な方法を, 加算器 (adder) を用いる場合について述べよ。

(2) このアルゴリズムを実現するデータパス (datapath) と制御回路 (control circuit) を設計する。データパスは図 1 のとおりであり, レジスタ A (5 ビット), レジスタ Q (4 ビット), レジスタ D (4 ビット), および 5 ビットの加減算器 (adder-subtractor) から構成される。

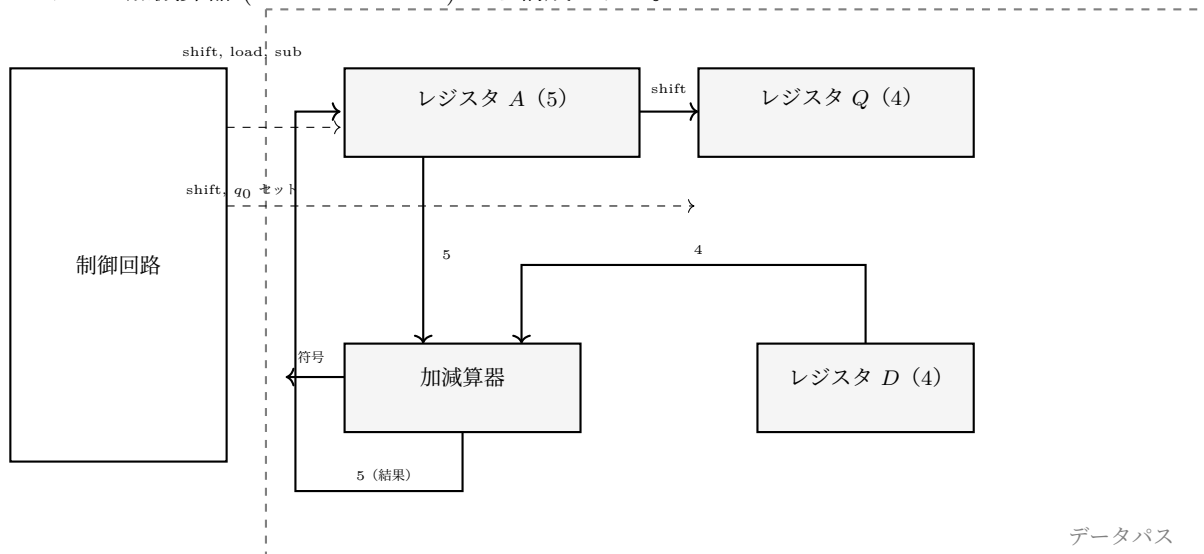


図 1 除算器のデータパス (実線：データ, 破線：制御信号)

(2-1) 制御回路を Moore 型同期式順序回路 (Moore synchronous sequential circuit) として設計する。状態は初期化状態 S_{INIT} , シフト状態 S_{SFT} , 減算状態 S_{SUB} , 判定・復元状態 S_{FIX} , 完了状態 S_{DONE} の 5 状態とす

る。繰り返し回数を数えるために 3 ビットのカウンタを用い、その値が 4 に達したら S_{DONE} へ遷移する。この制御回路の状態遷移図を、各状態における出力とともに示せ。

(2-2) 小問 (2-1) の状態のうち S_{SFT} , S_{SUB} , S_{FIX} の 3 状態のみに着目し、状態割り当てを $S_{\text{SFT}} = (y_1, y_0) = (0, 0)$, $S_{\text{SUB}} = (0, 1)$, $S_{\text{FIX}} = (1, 0)$ とする。カウンタの値が 4 に達したことを表す信号を c (達したとき 1) とし、 S_{FIX} からは $c = 0$ のとき S_{SFT} へ、 $c = 1$ のとき $S_{\text{DONE}} = (1, 1)$ へ遷移するものとする。二つの D フリップフロップ (D flip-flop) で実現するときの D_1 , D_0 の論理式の最簡積和形 (最小積和形) を、 c, y_1, y_0 を用いて求めよ。

(3) 設計の評価について以下の各小問に答えよ。

(3-1) この設計では、4 ビットの除算に少なくとも何クロックサイクルを要するかを答えよ。ただし S_{INIT} と S_{DONE} をそれぞれ 1 サイクルと数えるものとする。

(3-2) 除数 D が 0 である場合、このハードウェアはどのような動作をするかを述べよ。また、それを検出するために追加すべき回路を 1 行で述べよ。

7 【選択問題】数学解析と信号処理

配点：(1-1) 15 点, (1-2) 20 点, (1-3) 20 点, (2-1) 15 点, (2-2) 20 点, (2-3) 20 点, (2-4) 15 点

連続時間系と離散時間系のそれぞれについて以下の各問に答えよ。ただし導出の過程も示すこと。 j を虚数単位 (imaginary unit) とする。

(1) 図 1 に示す RLC 直列回路の過渡応答 (transient response) をラプラス変換 (Laplace transform) により求めたい。時刻 $t < 0$ において回路は静止しており、コンデンサの電荷およびインダクタの電流はいずれも 0 であった。時刻 $t = 0$ で大きさ E の直流電圧が印加される。

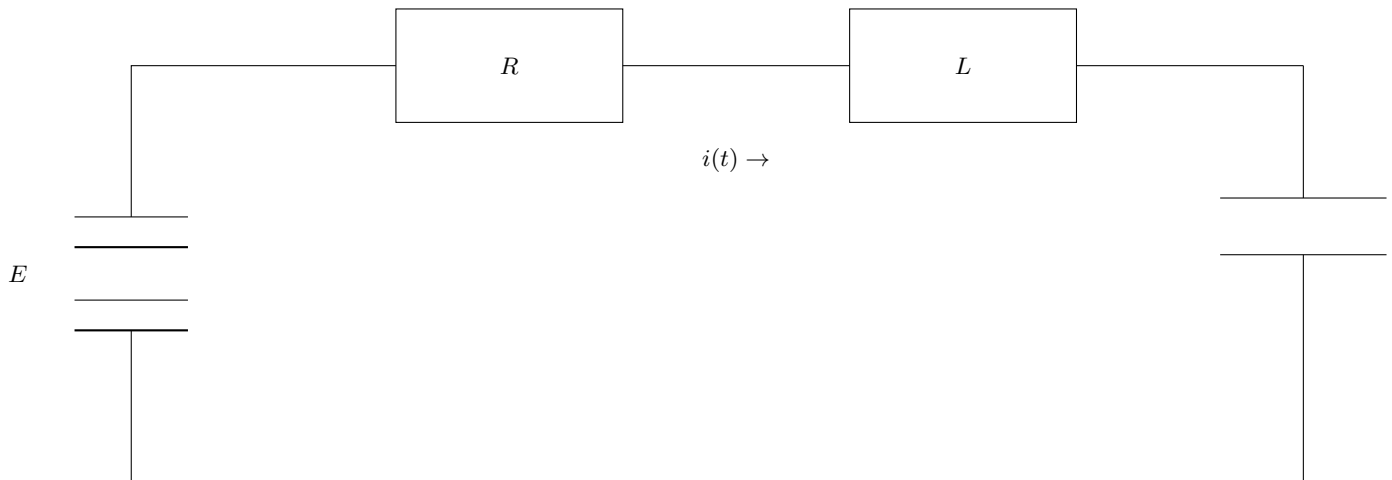


図 1 RLC 直列回路

(1-1) 回路に流れる電流を $i(t)$ とする。回路方程式をラプラス変換し、 $I(s) = \mathcal{L}[i(t)]$ を E, R, L, C, s を用いて表せ。

(1-2) $R = 3\Omega$, $L = 1\text{H}$, $C = 0.5\text{F}$, $E = 6\text{V}$ とする。このとき $I(s)$ を求め、部分分数分解 (partial fraction decomposition) により $i(t)$ ($t \geq 0$) を求めよ。

(1-3) 小問 (1-2) の $I(s)$ の極 (pole) をすべて求め、それらの位置に基づいてこの回路の応答が振動的 (oscillatory) であるか否かを判定せよ。また、 R の値を小さくしていったとき、どのような値を境に応答の性質が変わるかを、 R, L, C を用いた条件式で示せ。

(2) 離散時間系について以下の各小問に答えよ。 z 変換 (z -transform) を $X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$ で定義する。いま、因果的 (causal) な離散時間システムの伝達関数 (transfer function) が

$$H(z) = \frac{1 + z^{-1}}{1 - 0.5z^{-1}}$$

で与えられているとする。

(2-1) このシステムの極と零点 (zero) をすべて求めよ。また、極の位置に基づいてこのシステムが BIBO 安定 (bounded-input bounded-output stable) であるか否かを判定せよ。

(2-2) このシステムの入力 $x[n]$ と出力 $y[n]$ の関係を表す差分方程式 (difference equation) を示せ。さらに、インパルス応答 $h[n]$ を求めよ。(ヒント： $H(z)$ を $\frac{1}{1 - 0.5z^{-1}}$ と $\frac{z^{-1}}{1 - 0.5z^{-1}}$ の和に分けよ。)

(2-3) 周波数応答 (frequency response) $H(e^{j\omega})$ について、 $\omega = 0$ および $\omega = \pi$ における $|H(e^{j\omega})|$ の値をそれぞれ求めよ。また、このシステムが低域通過 (low-pass) と高域通過 (high-pass) のどちらの特性をもつかを、計算結果に基づいて答えよ。

(2-4) 小問 (2-3) で $\omega = \pi$ における値が特別な値となる理由を、零点の位置と単位円 (unit circle) の関係に基づいて 1, 2 行で説明せよ。