情報工学

予想問題　第5回　解答

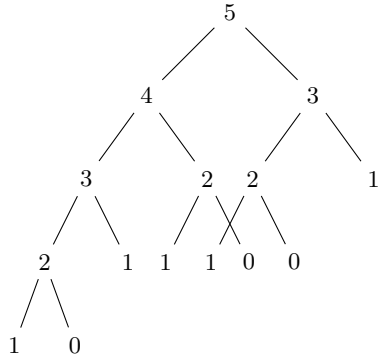
（証明・構成問題の重い年）

記述式の小問については解答例を示す。同じ内容が伝われば表現は問わない。

作成：Claude（Anthropic）

1 【必須問題】アルゴリズムとプログラミング

(1-1) `funcA(5)` の呼び出しの木は以下のとおり（括弧内は引数）。



各引数についての呼び出し回数は以下のとおり。

引数 n	5	4	3	2	1	0	合計
呼び出し回数	1	1	2	3	5	3	**15**

(1-2) `funcA(n)` は自分自身の 1 回の呼び出しに加えて，$n \geq 2$ のとき `funcA(n-1)` と `funcA(n-2)` を呼び出し，それぞれがさらに再帰的な呼び出しを生む。したがって

$$C(n) = C(n-1) + C(n-2) + 1 \qquad (n \geq 2)$$

が成り立つ。また $n \leq 1$ のときは再帰呼び出しを行わず自分自身の 1 回のみであるから $C(0) = C(1) = 1$ である。

次に $C(n) \geq F_n$ を n に関する数学的帰納法で示す。

基底段階：$C(0) = 1 \geq 0 = F_0$，$C(1) = 1 \geq 1 = F_1$ であり成り立つ。

帰納段階：$n \geq 2$ とし，$C(n-1) \geq F_{n-1}$ かつ $C(n-2) \geq F_{n-2}$ が成り立つと仮定する。このとき

$$C(n) = C(n-1) + C(n-2) + 1 \geq F_{n-1} + F_{n-2} + 1 > F_{n-1} + F_{n-2} = F_n.$$

よってすべての $n \geq 0$ について $C(n) \geq F_n$ が成り立つ。

F_n は n に対して指数的に増加する（$F_n \sim \varphi^n/\sqrt{5}$，$\varphi = (1+\sqrt{5})/2$）から，$C(n)$ も少なくとも指数的に増加する。関数呼び出し 1 回あたりの処理時間は定数であるから，`funcA` の時間計算量は n のいかなる多項式によっても上から抑えられない。

(2-1) (a) `memo[n] != -1`　(b) `funcB(n - 1) + funcB(n - 2)`

(2-2) 出力は以下のとおり。

```
5 15
5 9
```

（1 行目は `funcA(5)` の戻り値 $F_5 = 5$ と呼び出し回数 15，2 行目は `funcB(5)` の戻り値 5 と呼び出し回数 $2 \cdot 5 - 1 = 9$ である。）

(2-3) k を $2 \leq k \leq n$ を満たす整数とする。`memo[k]` が未計算（値が -1）の状態で `funcB(k)` が呼ばれると，15 行目で `memo[k]` に値が代入されるので，以後 `funcB(k)` が呼ばれても 14 行目で直ちに `return` し，再帰呼び出しは生じない。したがって「未計算の状態での呼び出し」は各 k について高々 1 回である。

`funcB(n)` を最初に呼び出すと，$k = n, n-1, \ldots, 2$ のそれぞれについて未計算の状態での呼び出しがちょうど 1 回ずつ生じ，そのたびに 2 回の再帰呼び出し（子）が発生する。すなわち子の呼び出しの総数は $2(n-1)$ である。これに最初の呼び出し 1 回を加えて，総呼び出し回数は

$$1 + 2(n-1) = 2n - 1.$$

(2-4) 例えば以下のように書ける。

```
int fib(int n) {
    int a = 0, b = 1, t;
    for (int i = 0; i < n; i++) {
        t = a + b;
        a = b;
        b = t;
    }
    return a;
}
```

ループが n 回まわり各反復の処理は定数時間であるから，時間計算量は $\mathbf{O(n)}$。使用する記憶領域は変数 a, b, t の 3 個のみで入力の大きさによらないから，空間計算量は $\mathbf{O(1)}$ である。

(3-1) funcA(n) は funcA(n-1) と funcA(n-2) を呼び出すが，深さが最大となるのは常に第 1 引数側（$n-1$）へ降り続ける経路である。この経路では引数が $n, n-1, n-2, \ldots, 1$ と 1 ずつ減少するので，再帰の最大の深さは $\mathbf{n}$ である（最初の呼び出しを深さ 1 と数える）。

記憶領域の差：再帰による実装では呼び出しごとに引数や戻り番地を保持する領域が呼び出しスタック上に積まれるため $O(n)$ の記憶領域を要する。一方ループによる実装は変数 3 個のみで済み $O(1)$ である。n が非常に大きい場合，再帰版はスタックオーバーフローを起こす危険がある。

(3-2) n に関する数学的帰納法で示す。

基底段階： $n = 1$ のとき，$M^1 = M = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ である。一方 $F_2 = 1$，$F_1 = 1$，$F_0 = 0$ であるから

$$\begin{pmatrix} F_2 & F_1 \\ F_1 & F_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

となり一致する。

帰納段階： $n \geq 1$ に対して主張が成り立つと仮定する。このとき

$$M^{n+1} = M^n M = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} F_{n+1} + F_n & F_{n+1} \\ F_n + F_{n-1} & F_n \end{pmatrix}.$$

漸化式 $F_{n+2} = F_{n+1} + F_n$ および $F_{n+1} = F_n + F_{n-1}$ を用いると

$$M^{n+1} = \begin{pmatrix} F_{n+2} & F_{n+1} \\ F_{n+1} & F_n \end{pmatrix}$$

となり，$n+1$ についても主張が成り立つ。

以上より，すべての $n \geq 1$ について $M^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$ である。$\boxtimes$

(3-3) 必要な演算回数のオーダは $\mathbf{O}(\log \mathbf{n})$ である。

理由：繰り返し二乗法では M^n を求めるのに，$M^2 = M \cdot M$，$M^4 = M^2 \cdot M^2$，$\ldots$ と二乗を繰り返す。1 回二乗するごとに指数が 2 倍になるので，M^{2^t} を得るまでに t 回の乗算で済み，$2^t \leq n$ より $t \leq \log_2 n$ である。n が 2 のべき乗でない場合も，n を 2 進表現したときの各ビットに対応する M^{2^t} を掛け合わせればよく，そのビット数は $\lfloor \log_2 n \rfloor + 1$ である。したがって全体で高々 $2\log_2 n$ 程度の行列乗算で計算でき，$O(\log n)$ となる。

2 【必須問題】計算機システムとシステムプログラム

(1-1) 参照列 R_1，$k = 3$ の場合。◯ はページフォルト，× はヒットを表し，括弧内は追い出したページを示す。

FIFO 方式（9 回）

参照	2	3	2	1	5	2	4	5	3	2	5	2
枠 1	2	2	2	2	5	5	5	5	3	3	3	3
枠 2	—	3	3	3	3	2	2	2	2	2	5	5
枠 3	—	—	—	1	1	1	4	4	4	4	4	2
判定	◯	◯	×	◯	◯(2)	◯(3)	◯(1)	×	◯(5)	×	◯(2)	◯(4)

LRU 方式（7 回）

参照	2	3	2	1	5	2	4	5	3	2	5	2
枠 1	2	2	2	2	2	2	2	2	3	3	3	3
枠 2	—	3	3	3	5	5	5	5	5	5	5	5
枠 3	—	—	—	1	1	1	4	4	4	2	2	2
判定	◯	◯	×	◯	◯(3)	×	◯(1)	×	◯(2)	◯(4)	×	×

OPT 方式（6 回）

参照	2	3	2	1	5	2	4	5	3	2	5	2
枠 1	2	2	2	2	2	2	4	4	4	2	2	2
枠 2	—	3	3	3	3	3	3	3	3	3	3	3
枠 3	—	—	—	1	5	5	5	5	5	5	5	5
判定	◯	◯	×	◯	◯(1)	×	◯(2)	×	×	◯(4)	×	×

まとめると，**FIFO：9 回，LRU：7 回，OPT：6 回**である。

(1-2) OPT は，メモリ上のページのうち**以後最も長く参照されない（あるいは二度と参照されない）ページ**を選んで追い出す。

実装できない理由：将来のページ参照列をあらかじめ知っている必要があるが，実際のシステムでは未来の参照を知ることはできないため。（他方式の性能を評価する理論的な下限として用いられる。）

(1-3) 小問 (1) の結果は FIFO 9 回，LRU 7 回であり，**LRU の方が 2 回少ない**。

理由：プログラムのメモリ参照には，最近参照した領域を近い将来ふたたび参照しやすいという時間的局所性 (temporal locality) がある。LRU は「最後に参照された時点が最も古いページ」を追い出すため，この局所性を直接利用して，今後使われる可能性の高いページをメモリ上に残すことができる。一方 FIFO は読み込まれた順序のみを基準とするため，頻繁に参照されているページであっても古くから存在するというだけで追い出してしまい，直後に再読み込みが必要になることがある。

(1-4) 参照列 $R_2 = 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5$ に FIFO を適用する。

$k = 3$ **の場合（9 回）**

参照	1	2	3	4	1	2	5	1	2	3	4	5
枠 1	1	1	1	4	4	4	5	5	5	5	5	5
枠 2	—	2	2	2	1	1	1	1	1	3	3	3
枠 3	—	—	3	3	3	2	2	2	2	2	4	4
判定	◯	◯	◯	◯(1)	◯(2)	◯(3)	◯(4)	×	×	◯(1)	◯(2)	×

$k = 4$ **の場合（10 回）**

参照	1	2	3	4	1	2	5	1	2	3	4	5
枠 1	1	1	1	1	1	1	5	5	5	5	4	4
枠 2	—	2	2	2	2	2	2	1	1	1	1	5
枠 3	—	—	3	3	3	3	3	3	2	2	2	2
枠 4	—	—	—	4	4	4	4	4	4	3	3	3
判定	◯	◯	◯	◯	×	×	◯(1)	◯(2)	◯(3)	◯(4)	◯(5)	◯(1)

ページ枠を 3 個から 4 個に増やしたにもかかわらず，ページフォルトが 9 回から **10 回に増加**している。この現象を **Belady の異常** (Belady's anomaly) という。

(1-5)　LRU では，任意の時点でメモリ上に存在するのは「直近に参照された k 個のページ」である。したがって枠数 k のときのページ集合は，枠数 $k+1$ のときのページ集合の**部分集合**となる（この性質をスタック性質 (stack property) という）。

ある参照がヒットするのは，そのページがメモリ上に存在するときである。上の包含関係より，枠数 k でヒットする参照は枠数 $k+1$ でも必ずヒットする。ゆえに枠を増やしてフォルト回数が増えることはあり得ず，Belady の異常は発生しない。

(1-6)　参照列の例：**2**, **1**, **2**, **3**, **4**, **1**（$k=3$）

LRU 方式（5 回）

参照	2	1	2	3	4	1
枠 1	2	2	2	2	2	1
枠 2	—	1	1	1	4	4
枠 3	—	—	—	3	3	3
判定	○	○	×	○	○(1)	○(2)

FIFO 方式（4 回）

参照	2	1	2	3	4	1
枠 1	2	2	2	2	4	4
枠 2	—	1	1	1	1	1
枠 3	—	—	—	3	3	3
判定	○	○	×	○	○(2)	×

説明：5 番目の参照 4 の時点で，LRU は最後に参照された時点が最も古いページとして 1 を追い出すが，直後に 1 が参照されるためフォルトとなる。一方 FIFO は最初に読み込まれたページとして 2 を追い出すので 1 がメモリ上に残り，最後の参照 1 がヒットとなる。このように LRU の前提である局所性が成り立たない参照列では，FIFO の方がフォルト回数が少なくなり得る。

(1-7)　各ページ枠に参照ビットを 1 ビット持たせる。ページが参照されるとハードウェアがそのビットを 1 にセットする。ページ枠を円環状に並べ，追い出し対象を指す針（ポインタ）を保持する。

置換が必要になったとき，針が指す枠の参照ビットを調べる。ビットが 1 であれば「最近使われた」とみなして追い出さず，そのビットを 0 にクリアして針を次の枠へ進める（セカンドチャンスを与える）。ビットが 0 の枠が見つかったら，そのページを追い出して新しいページを格納し，針を次へ進める。

これにより，最近参照されたページを優先的に残すという LRU の趣旨を，参照時刻の順序を厳密に管理することなく 1 ビットの情報と巡回走査のみで近似的に実現できる。

(2-1)　1 回転に要する時間は

$$\frac{60}{7200}\ \text{秒} = \frac{60000}{7200}\ \text{ms} = 8.333\ \text{ms}.$$

平均回転待ち時間は半回転分であるから

$$\frac{8.333}{2} = \mathbf{4.17}\ [\mathbf{ms}].$$

理由：目的のセクタがヘッドの直下に来るまでの待ち時間は 0 から 1 回転分までの間で一様に分布すると考えられるため，その平均は半回転分となる。

(2-2)　1 トラックに 500 セクタがあり 1 回転で全セクタが通過するので，1 セクタの転送時間は

$$\frac{8.333}{500} = \mathbf{0.0167}\ [\mathbf{ms}].$$

ランダムな位置にある 1 セクタの平均アクセス時間は，平均シーク時間・平均回転待ち時間・転送時間の和であるから

$$8.0 + 4.17 + 0.0167 = \mathbf{12.2}\ [\mathbf{ms}].$$

(2-3) 4 KiB = 4096 バイトは 4096/512 = 8 セクタ分である。

連続配置の場合：シークと回転待ちは最初の 1 回のみ生じ，8 セクタ分を連続して転送する。

$$8.0 + 4.17 + 8 \times 0.0167 = 8.0 + 4.17 + 0.133 = \mathbf{12.3}\ [\mathbf{ms}].$$

8 か所に分散している場合：各セクタごとにシークと回転待ちが生じる。

$$8 \times 12.2 = \mathbf{97.5}\ [\mathbf{ms}].$$

比較と理由：分散配置は連続配置の約 8 倍の時間を要する。これは，実際のデータ転送時間（0.0167 ms）に比べてシーク時間（8.0 ms）と回転待ち時間（4.17 ms）が 3 桁近く大きく，アクセス時間の大部分がヘッドの位置決めに費やされるためである。したがってファイルシステムは，位置決めの回数を減らすためデータをなるべく連続した領域に配置しようとする。

(2-4) ページフォルト 1 回あたりの処理時間を小問 (2-2) で求めた $12.2\,\mathrm{ms} = 1.22 \times 10^7\,\mathrm{ns}$ とする。ページフォルト率を p とすると

$$\mathrm{EAT} = (1-p) \times 100 + p \times \left(1.22 \times 10^7 + 100\right) \approx 100 + p \times 1.22 \times 10^7 \quad [\mathrm{ns}].$$

$p = 5.0 \times 10^{-6}$ を代入して

$$\mathrm{EAT} = 100 + 5.0 \times 10^{-6} \times 1.22 \times 10^7 = 100 + 61 = \mathbf{161}\ [\mathbf{ns}].$$

すなわちページフォルトがなければ 100 ns で済むところ，20 万回に 1 回程度のフォルトだけで実効アクセス時間が 1.6 倍に悪化する。

多重プログラミング度を上げすぎるとページフォルト率が急増してシステム全体の処理能力が著しく低下する現象を**スラッシング** (thrashing) という。

4 【選択問題】計算理論

---

(1) 入力の a に対して 0 を，b に対して 1 をスタックに積む。前半を積み終えた時点（中央）で照合の状態 q_1 へ移り，以降はスタックトップと入力を照合しながら取り去る。スタックが Z のみになったら受理状態 q_2 へ移る。中央の位置は入力から判別できないため，$q_0 \to q_1$ の遷移を非決定的に行う。

q_0 **の自己ループ（積み込み）：**	$(a,Z)/0Z$, $(a,0)/00$, $(a,1)/01$
	$(b,Z)/1Z$, $(b,0)/10$, $(b,1)/11$
$q_0 \to q_1$ **（偶数長：中央を推測）：**	$(\varepsilon,Z)/Z$, $(\varepsilon,0)/0$, $(\varepsilon,1)/1$
$q_0 \to q_1$ **（奇数長：中央の 1 文字を読み捨て）：**	$(a,Z)/Z$, $(a,0)/0$, $(a,1)/1$
	$(b,Z)/Z$, $(b,0)/0$, $(b,1)/1$
q_1 **の自己ループ（照合）：**	$(a,0)/\varepsilon$, $(b,1)/\varepsilon$
$q_1 \to q_2$ **（受理）：**	$(\varepsilon,Z)/Z$

非決定性を用いている箇所：q_0 において，「積み込みを続ける」か「q_1 へ移って照合を始める」かの選択，および q_1 へ移る際に「入力を読まない（偶数長）」か「1 文字読み捨てる（奇数長）」かの選択である。すなわち文字列の中央がどこであるかの推測に非決定性を用いている。

(2) L_{pal} が正則言語であると仮定する。反復補題により，条件を満たす正の整数 n が存在する。

ここで $v = a^n b a^n$ とすると，v は回文であるから $v \in L_{\mathrm{pal}}$ であり，$|v| = 2n+1 \geq n$ である。よって $v = xyz$（$|xy| \leq n$，$|y| \geq 1$）と分解でき，任意の $k \geq 0$ に対して $xy^k z \in L_{\mathrm{pal}}$ が成り立つ。

$|xy| \leq n$ であるから，x と y はいずれも v の先頭の a^n の部分に含まれる。したがって $y = a^m$（$1 \leq m \leq n$）と書ける。

ここで $k = 0$ をとると

$$xy^0z = xz = a^{n-m}\, b\, a^n$$

となる。この文字列は b の前に a が $n-m$ 個，後ろに a が n 個並んでおり，$m \geq 1$ より $n - m \neq n$ である。したがって前から読んだ場合と後ろから読んだ場合が一致せず，回文ではない。すなわち $xz \notin L_{\mathrm{pal}}$ であり，反復補題の主張に矛盾する。

よって仮定が誤りであり，L_{pal} は正則言語ではない。☒

(3-1) $w = a^p b^p$ とおくと

$$ww = a^p b^p a^p b^p = s$$

となるから，$s \in L_{ww}$ である。また $|s| = 4p$ であり，$p \geq 1$ より $4p \geq p$ が成り立つので $|s| \geq p$ である。

(3-2) 空欄の答えは以下のとおり。

(ア) 奇数

（L_{ww} に属する文字列は ww の形をしているので長さは必ず偶数である。$|s'| = 4p - |vy|$ において $|vy|$ が奇数であれば $|s'|$ は奇数となり，L_{ww} に属し得ない。）

(イ) 高々二つの隣接するブロックにまたがる（三つ以上のブロックにまたがることはない）

（各ブロックの長さは p であり $|vxy| \leq p$ であるから，vxy が三つのブロックにまたがるとすると中央のブロック全体（長さ p）を含むうえに両隣にもはみ出すことになり，$|vxy| > p$ となって条件 3 に反する。）

(ウ) 以下のような記述であればよい。

vxy は前半の $a^p b^p$ に収まるので，v と y から取り除かれるのは第 1 ブロックの a と第 2 ブロックの b のみである。取り除かれる a の個数を α，b の個数を β とすると $\alpha + \beta = |vy| > 0$ であり，

$$s' = a^{p-\alpha}\, b^{p-\beta}\, a^p\, b^p$$

となる。$d = |vy|/2 > 0$ とおくと $|s'| = 4p - 2d$ であるから，$s' = w'w'$ と書けるとすれば $|w'| = 2p - d < 2p$ である。

後半の w' は s' の末尾 $2p - d$ 文字であり，s' の末尾 $2p$ 文字は変化していない $a^p b^p$ であるから，後半の $w' = a^{p-d} b^p$ となり，末尾は b である。一方，前半の w' は s' の先頭 $2p - d$ 文字であり，第 1・第 2 ブロックの残り $a^{p-\alpha} b^{p-\beta}$（長さ $2p - 2d$）に続いて第 3 ブロックの a が d 個並ぶので，末尾は a である（$d > 0$）。

前半と後半の末尾の文字が異なるので $s' \neq w'w'$ であり，$s' \notin L_{ww}$ である。

(エ) 以下のような記述であればよい。

vxy は中央の $b^p a^p$ の境界をまたぐので，v と y から取り除かれるのは第 2 ブロックの b と第 3 ブロックの a のみであり，第 1 ブロックの a^p と第 4 ブロックの b^p は変化しない。取り除かれる b の個数を β，a の個数を α とすると

$$s' = a^p\, b^{p-\beta}\, a^{p-\alpha}\, b^p$$

となる。$d = |vy|/2 > 0$ とおくと $|w'| = 2p - d < 2p$ である。

s' の先頭は変化していない a^p であり，$|w'| = 2p - d > p$（$d \leq p/2 < p$ による）であるから，前半の w' は a を p 個並べた接頭辞をもつ。同様に s' の末尾は変化していない b^p であるから，後半の w' は b を p 個並べた接尾辞をもつ。

同一の文字列 w' が長さ p の a のみの接頭辞と長さ p の b のみの接尾辞をともにもつには $|w'| \geq 2p$ でなければならない。しかし $|w'| = 2p - d < 2p$ であるから矛盾する。よって $s' \notin L_{ww}$ である。

(4) まず $L_1 = \{a^n b^n c^m \mid n, m \geq 0\}$ は，例えば

$$S \to XY, \qquad X \to aXb \mid \varepsilon, \qquad Y \to cY \mid \varepsilon$$

という文脈自由文法で生成されるので文脈自由言語である。同様に $L_2 = \{a^m b^n c^n \mid n, m \geq 0\}$ も

$$S \to XY, \qquad X \to aX \mid \varepsilon, \qquad Y \to bYc \mid \varepsilon$$

により生成されるので文脈自由言語である。

一方，L_1 に属する語は a と b の個数が等しく，L_2 に属する語は b と c の個数が等しい。両方に属する語は a, b, c がこの順に並び，かつ三つの個数がすべて等しいものであるから

$$L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}.$$

これは文脈自由言語でないことが証明済みである。すなわち，文脈自由言語 L_1, L_2 の積集合が文脈自由言語とならない例が存在する。したがって文脈自由言語は積集合について閉じていない。$\boxtimes$

6 【選択問題】電子回路と論理設計

(1-1)

$$E = \overline{(a_1 \oplus b_1)} \cdot \overline{(a_0 \oplus b_0)}$$

理由：2 ビットの数 A と B が等しいことは，上位ビットどうしが一致し（$a_1 = b_1$），かつ下位ビットどうしが一致する（$a_0 = b_0$）ことと同値である。$a \oplus b$ は $a \neq b$ のとき 1 となるから，その否定 $\overline{a \oplus b}$（XNOR）は $a = b$ のとき 1 となる。両ビットについての一致条件の論理積をとれば $A = B$ の判定となる。

(1-2) $A > B$ となる (A, B) の組は，$(A, B) = (1,0), (2,0), (2,1), (3,0), (3,1), (3,2)$ の 6 通りである。これを $a_1 a_0 b_1 b_0$ の最小項で表すと

$$m(0100),\ m(1000),\ m(1001),\ m(1100),\ m(1101),\ m(1110)$$

すなわち $m(4), m(8), m(9), m(12), m(13), m(14)$ である。

カルノー図上でこれらをまとめると，次の三つのグループが得られる。

- $a_1 = 1, b_1 = 0$ の 4 マス（$m(8), m(9), m(12), m(13)$）$\Rightarrow a_1 \overline{b_1}$
- $a_0 = 1, b_1 = 0, b_0 = 0$ の 2 マス（$m(4), m(12)$）$\Rightarrow a_0 \overline{b_1}\,\overline{b_0}$
- $a_1 = 1, a_0 = 1, b_0 = 0$ の 2 マス（$m(12), m(14)$）$\Rightarrow a_1 a_0 \overline{b_0}$

よって最簡積和形は

$$G = a_1 \overline{b_1} + a_0 \overline{b_1}\,\overline{b_0} + a_1 a_0 \overline{b_0}.$$

（意味：上位ビットで勝つ場合，上位が等しく 0 で下位で勝つ場合，上位が等しく 1 で下位で勝つ場合，に対応している。）

(2-1) 与えられた接続 $D_2 = \overline{y_0}$，$D_1 = y_2$，$D_0 = y_1$ に従って状態遷移を追うと，

$$000 \to 100 \to 110 \to 111 \to 011 \to 001 \to 000$$

となり，6 状態を巡回して初期状態に戻る。

このカウンタの名称は**ジョンソンカウンタ**（ねじれリングカウンタ，Johnson counter / twisted ring counter）である。

(2-2) 8 通りの状態のうち，上記の巡回に含まれないのは **010** と **101** の二つである。

これらを初期状態としたときの遷移を調べると

$$010 \to 101 \to 010 \to 101 \to \cdots$$

となり，この二つの状態の間を永久に往復し続け，正規の 6 状態の巡回に戻ることができない。

生じる問題の名称：**(B) ロックアウト (lockout)**

対策の例：電源投入時にリセット信号を与えて必ず正規の巡回に含まれる状態（例えば 000）に初期化する。あるいは，未使用状態を検出して強制的に正規の系列へ遷移させる自己修正 (self-correcting) 論理を付加する（例えば $D_2 = \overline{y_0}$ を $D_2 = \overline{y_0} \cdot (\overline{y_1} + y_2 \cdot \ldots)$ のように補正する構成をとる）。

(2-3) 通常の 2 進カウンタでは，例えば $011 \to 100$ のように複数のフリップフロップが同時に値を変える遷移がある。各フリップフロップの遅延には差があるため，遷移の途中で意図しない中間的な状態（この例では 001 や 111 など）が一瞬現れることがある。この状態を AND ゲートなどでデコードしていると，出力に短いパルス状の誤り，すなわちグリッチが生じる。

ジョンソンカウンタでは遷移ごとに変化するフリップフロップが常に一つだけであるため，遷移の途中に現れる状態は遷移前か遷移後のいずれかに限られ，意図しない中間状態が生じない。したがってデコード出力にグリッチが発生せず，デコード結果をそのまま制御信号として安全に利用できる。

(2-4) n 個の D フリップフロップを用いたジョンソンカウンタの正規の巡回に含まれる状態数は **2n** 個である。（$n = 3$ のとき $2 \times 3 = 6$ となり小問 (2-1) の結果と一致する。すべて 0 の状態から 1 が左端より 1 個ずつ入って全部 1 になるまでが n 状態，そこから 0 が 1 個ずつ入って全部 0 に戻るまでが n 状態である。）

短所：n ビットの 2 進カウンタは 2^n 個の状態を表現できるのに対し，ジョンソンカウンタは $2n$ 個しか使用しない。例えば 16 状態を得るには 2 進カウンタなら 4 個で済むが，ジョンソンカウンタでは 8 個のフリップフロップが必要となる。すなわち**同じ状態数を得るのに多くのフリップフロップを要し，状態の利用効率が低い**。

(2-1)

$$E_4 = E_H \cdot E_L, \qquad G_4 = G_H + E_H \cdot G_L.$$

G_4 の説明：4 ビットの大小比較では，まず上位 2 ビットどうしを比較する。上位で A の方が大きければ（$G_H = 1$），下位ビットの値によらず $A > B$ が確定する。上位が等しい場合（$E_H = 1$）に限り，下位 2 ビットの比較結果が全体の大小を決める。したがって「上位で勝つ」か「上位が引き分けで下位で勝つ」かの論理和となる。上位が小さい場合（$G_H = E_H = 0$）は下位を見るまでもなく $A < B$ であり，どちらの項も 0 となって正しく $G_4 = 0$ となる。

E_4 については，4 ビットすべてが一致することは上位 2 ビットと下位 2 ビットの双方が一致することと同値であるから，論理積となる。

(2-2) $A = 1001_2$，$B = 1010_2$ より，上位 2 ビットは $(a_3, a_2) = (1, 0)$，$(b_3, b_2) = (1, 0)$ であり，下位 2 ビットは $(a_1, a_0) = (0, 1)$，$(b_1, b_0) = (1, 0)$ である。

上位：$A_H = 10_2 = 2$，$B_H = 10_2 = 2$ であるから $A_H = B_H$ より

$$E_H = 1, \qquad G_H = 0.$$

下位：$A_L = 01_2 = 1$，$B_L = 10_2 = 2$ であるから $A_L < B_L$ より

$$E_L = 0, \qquad G_L = 0.$$

これを小問 (3-1) の式に代入して

$$E_4 = E_H \cdot E_L = 1 \cdot 0 = 0, \qquad G_4 = G_H + E_H \cdot G_L = 0 + 1 \cdot 0 = 0.$$

一方 $A = 1001_2 = 9$，$B = 1010_2 = 10$ であるから $A < B$ であり，$E_4 = 0$（等しくない）かつ $G_4 = 0$（$A > B$ でない）という結果と一致する。

(3-1)

$$E_4 = E_H \cdot E_L, \qquad G_4 = G_H + E_H \cdot G_L.$$

G_4 の説明：4 ビットの大小比較では，まず上位 2 ビットどうしを比較する。上位で A の方が大きければ（$G_H = 1$），下位ビットの値によらず $A > B$ が確定する。上位が等しい場合（$E_H = 1$）に限り，下位 2 ビットの比較結果が全体の大小を決める。したがって「上位で勝つ」か「上位が引き分けで下位で勝つ」かの論理和となる。上位が小さい場合（$G_H = E_H = 0$）は下位を見るまでもなく $A < B$ であり，どちらの項も 0 となって正しく $G_4 = 0$ となる。

E_4 については，4 ビットすべてが一致することは上位 2 ビットと下位 2 ビットの双方が一致することと同値であるから，論理積となる。

(3-2) $A = 1001_2$，$B = 1010_2$ より，上位 2 ビットは $(a_3, a_2) = (1, 0)$，$(b_3, b_2) = (1, 0)$，下位 2 ビットは $(a_1, a_0) = (0, 1)$，$(b_1, b_0) = (1, 0)$ である。

上位：$A_H = 10_2 = 2$，$B_H = 10_2 = 2$ で等しいから

$$E_H = 1, \qquad G_H = 0.$$

下位：$A_L = 01_2 = 1$，$B_L = 10_2 = 2$ で $A_L < B_L$ であるから

$$E_L = 0, \qquad G_L = 0.$$

小問 (3-1) の式に代入して

$$E_4 = E_H \cdot E_L = 1 \cdot 0 = 0, \qquad G_4 = G_H + E_H \cdot G_L = 0 + 1 \cdot 0 = 0.$$

一方 $A = 1001_2 = 9$，$B = 1010_2 = 10$ であるから $A < B$ であり，$E_4 = 0$（等しくない）かつ $G_4 = 0$（$A > B$ でない）という結果と一致する。

7 【選択問題】数学解析と信号処理

(1-1) $f(x) = |x|$ は偶関数 (even function) であり，$\sin nx$ は奇関数であるから，その積 $f(x)\sin nx$ は奇関数となる。奇関数を原点対称な区間で積分すると 0 になるので，すべての $n \geq 1$ について

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin nx\, dx = 0$$

である。

a_0 については

$$a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} |x|\, dx = \frac{2}{\pi}\int_0^{\pi} x\, dx = \frac{2}{\pi}\cdot\frac{\pi^2}{2} = \pi.$$

a_n（$n \geq 1$）については，偶関数性より

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} |x|\cos nx\, dx = \frac{2}{\pi}\int_0^{\pi} x\cos nx\, dx.$$

部分積分により

$$\int_0^{\pi} x\cos nx\, dx = \left[\frac{x\sin nx}{n}\right]_0^{\pi} - \frac{1}{n}\int_0^{\pi}\sin nx\, dx = 0 + \frac{1}{n}\left[\frac{\cos nx}{n}\right]_0^{\pi} = \frac{\cos n\pi - 1}{n^2}.$$

よって

$$a_n = \frac{2}{\pi}\cdot\frac{\cos n\pi - 1}{n^2} = \frac{2\left((-1)^n - 1\right)}{\pi n^2}.$$

n が偶数のとき $a_n = 0$，n が奇数のとき $a_n = -\dfrac{4}{\pi n^2}$ である。

したがって $n = 2k+1$（$k \geq 0$）とおいて

$$f(x) = \frac{\pi}{2} - \frac{4}{\pi}\sum_{k=0}^{\infty} \frac{\cos\bigl((2k+1)x\bigr)}{(2k+1)^2}.$$

(2-1) $x = 0$ を代入すると $f(0) = |0| = 0$ であり，$\cos 0 = 1$ であるから

$$0 = \frac{\pi}{2} - \frac{4}{\pi}\sum_{k=0}^{\infty} \frac{1}{(2k+1)^2}.$$

よって

$$\sum_{k=0}^{\infty} \frac{1}{(2k+1)^2} = \frac{\pi}{4}\cdot\frac{\pi}{2} = \frac{\pi^2}{8}.$$

(2-2) $S = \displaystyle\sum_{n=1}^{\infty} \frac{1}{n^2}$ を n が奇数の項と偶数の項に分ける。

奇数の項の和は，小問 (2-1) より

$$\sum_{k=0}^{\infty} \frac{1}{(2k+1)^2} = \frac{\pi^2}{8}.$$

偶数の項は $n = 2m$（$m \geq 1$）と書けるから

$$\sum_{m=1}^{\infty} \frac{1}{(2m)^2} = \sum_{m=1}^{\infty} \frac{1}{4m^2} = \frac{1}{4}\sum_{m=1}^{\infty} \frac{1}{m^2} = \frac{1}{4}S.$$

すなわち偶数の項の和は S 自身の $1/4$ で表せる。したがって

$$S = \frac{\pi^2}{8} + \frac{1}{4}S \quad\Longrightarrow\quad \frac{3}{4}S = \frac{\pi^2}{8} \quad\Longrightarrow\quad S = \frac{4}{3}\cdot\frac{\pi^2}{8} = \frac{\pi^2}{6}.$$

よって $\displaystyle\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$ である。

(2-3) パーセバルの等式の左辺を計算すると

$$\frac{1}{\pi}\int_{-\pi}^{\pi} f(x)^2 dx = \frac{1}{\pi}\int_{-\pi}^{\pi} x^2 dx = \frac{2}{\pi}\cdot\frac{\pi^3}{3} = \frac{2\pi^2}{3}.$$

右辺は，$b_n = 0$ および $a_0 = \pi$，$a_{2k+1} = -\dfrac{4}{\pi(2k+1)^2}$（偶数次は 0）より

$$\frac{a_0^2}{2} + \sum_{n=1}^{\infty} a_n^2 = \frac{\pi^2}{2} + \sum_{k=0}^{\infty} \frac{16}{\pi^2(2k+1)^4} = \frac{\pi^2}{2} + \frac{16}{\pi^2}\sum_{k=0}^{\infty} \frac{1}{(2k+1)^4}.$$

両者を等置して

$$\frac{16}{\pi^2}\sum_{k=0}^{\infty} \frac{1}{(2k+1)^4} = \frac{2\pi^2}{3} - \frac{\pi^2}{2} = \frac{4\pi^2 - 3\pi^2}{6} = \frac{\pi^2}{6}.$$

よって

$$\sum_{k=0}^{\infty} \frac{1}{(2k+1)^4} = \frac{\pi^2}{6}\cdot\frac{\pi^2}{16} = \frac{\pi^4}{96}.$$

(1-2) 複素形と実数形の係数の関係は以下のとおり。

$$c_0 = \frac{a_0}{2}, \qquad c_n = \frac{a_n - jb_n}{2}\ \ (n > 0), \qquad c_n = \frac{a_{|n|} + jb_{|n|}}{2}\ \ (n < 0).$$

（$e^{jnx} = \cos nx + j\sin nx$ を代入して $\cos nx = \frac{e^{jnx}+e^{-jnx}}{2}$，$\sin nx = \frac{e^{jnx}-e^{-jnx}}{2j}$ を用いれば導かれる。実数値関数では $c_{-n} = \overline{c_n}$ が成り立つ。）

$f(x) = |x|$ では $b_n = 0$ であるから $c_n = \dfrac{a_{|n|}}{2}$ となり，

$$c_0 = \frac{\pi}{2}, \qquad c_n = \begin{cases} -\dfrac{2}{\pi n^2} & (n \text{ が奇数}) \\ 0 & (n \text{ が偶数}, n \neq 0) \end{cases}$$

である（$n < 0$ の場合も $|n|$ が奇数なら同じ値をとる）。

(2-4) 部分和は

$$S_1(x) = \frac{\pi}{2} - \frac{4}{\pi}\cos x, \qquad S_2(x) = \frac{\pi}{2} - \frac{4}{\pi}\left(\cos x + \frac{\cos 3x}{9}\right).$$

$x = 0$ を代入すると

$$S_1(0) = \frac{\pi}{2} - \frac{4}{\pi} \approx 1.5708 - 1.2732 = 0.2976,$$

$$S_2(0) = \frac{\pi}{2} - \frac{4}{\pi}\left(1 + \frac{1}{9}\right) \approx 1.5708 - 1.4147 = 0.1561.$$

真の値は $f(0) = |0| = 0$ である。誤差は S_1 で約 0.298，S_2 で約 0.156 と約半分に減少しており，項数を増やすことで近似が改善されることが確認できる。

(2-5)　（前半：不等式が成り立つ理由）

$f(x) - S_N(x)$ は打ち切った残りの項の和であるから

$$f(x) - S_N(x) = -\frac{4}{\pi}\sum_{k=N}^{\infty} \frac{\cos\big((2k+1)x\big)}{(2k+1)^2}.$$

任意の x について $|\cos\theta| \le 1$ であるから，三角不等式により

$$\big|f(x) - S_N(x)\big| \le \frac{4}{\pi}\sum_{k=N}^{\infty} \frac{\big|\cos((2k+1)x)\big|}{(2k+1)^2} \le \frac{4}{\pi}\sum_{k=N}^{\infty} \frac{1}{(2k+1)^2}$$

が成り立つ。（右辺は x によらないので，一様な評価になっている。）

（後半：和の積分による評価）

関数 $g(t) = \dfrac{1}{(2t+1)^2}$ は $t > -1/2$ で単調減少である。したがって $k \ge N$ に対して

$$\frac{1}{(2k+1)^2} \le \int_{k-1}^{k} \frac{dt}{(2t+1)^2}$$

が成り立つ。両辺を $k = N$ から ∞ まで加えると

$$\sum_{k=N}^{\infty} \frac{1}{(2k+1)^2} \le \int_{N-1}^{\infty} \frac{dt}{(2t+1)^2} = \left[-\frac{1}{2(2t+1)}\right]_{N-1}^{\infty} = \frac{1}{2\big(2(N-1)+1\big)} = \frac{1}{2(2N-1)}.$$

（必要な項数）

以上より，すべての x について

$$\big|f(x) - S_N(x)\big| \le \frac{4}{\pi}\cdot\frac{1}{2(2N-1)} = \frac{2}{\pi(2N-1)}.$$

これを 0.01 以下にするには

$$\frac{2}{\pi(2N-1)} \le 0.01 \quad\Longleftrightarrow\quad 2N - 1 \ge \frac{200}{\pi} \approx 63.66 \quad\Longleftrightarrow\quad N \ge 32.3$$

であるから，$\mathbf{N = 33}$ とすれば十分である。

（検算：$N = 33$ のとき上界は $2/(\pi \cdot 65) \approx 0.0098 \leq 0.01$ であり，実際に $x = 0$ での誤差を数値計算すると約 0.0097 となって条件を満たす。）

(2-6) ギブス現象は，フーリエ級数の部分和が関数の**不連続点**の近傍で真の値を越えて振動し，部分和の項数を増やしてもその行き過ぎの大きさが一定の割合で残る現象である。

$f(x) = |x|$ は $-\pi \leq x \leq \pi$ で連続であり，かつ周期的に接続した端点 $x = \pm\pi$ においても $f(\pi) = f(-\pi) = \pi$ となって値が一致するため，実数全体で連続な関数となる。不連続点が存在しないので，ギブス現象が生じる原因そのものがない。実際，この級数の係数は $1/(2k+1)^2$ の速さで減衰し，フーリエ級数は f に一様収束する。（2020 年度に出題された関数のように不連続点をもつ場合には，その点でギブス現象が現れる。）

(3-1) 定義式に $N = 4$，$x[0] = x[1] = 1$，$x[2] = x[3] = 0$ を代入する。$e^{-j\frac{2\pi}{4}kn} = e^{-j\frac{\pi}{2}kn}$ であるから

$$X[k] = \sum_{n=0}^{3} x[n] e^{-j\frac{\pi}{2}kn} = 1 + e^{-j\frac{\pi}{2}k}.$$

したがって

$$X[0] = 1 + e^{0} = \mathbf{2},$$

$$X[1] = 1 + e^{-j\pi/2} = 1 + (-j) = \mathbf{1 - j},$$

$$X[2] = 1 + e^{-j\pi} = 1 + (-1) = \mathbf{0},$$

$$X[3] = 1 + e^{-j3\pi/2} = 1 + j = \mathbf{1 + j}.$$

(3-2) $X[1] = 1 - j$，$X[3] = 1 + j$ であるから，両者は互いに**複素共役**の関係にある。すなわち $X[3] = \overline{X[1]}$ である。

一般に，$x[n]$ が実数値信号であるとき

$$X[N-k] = \overline{X[k]}$$

が成り立つ。

理由：定義式より

$$X[N-k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}(N-k)n} = \sum_{n=0}^{N-1} x[n] e^{-j2\pi n} e^{j\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} x[n] e^{j\frac{2\pi}{N}kn}$$

（$e^{-j2\pi n} = 1$ による）となる。$x[n]$ が実数であれば，これは $X[k]$ の各項の複素共役を足し合わせたものに等しく，$\overline{X[k]}$ となる。

（この性質により，実数値信号の DFT では $k = 0$ から $N/2$ までを求めれば残りは共役として得られ，計算量と記憶量を半減できる。）

(3-3) 本質的な違いは，得られる周波数成分の**個数が有限か無限か**である。

連続時間の周期関数に対するフーリエ級数展開では，基本周波数の整数倍の周波数成分が無限個（$n = 1, 2, 3, \ldots$）現れ，級数は無限和となる。一方，長さ N の離散時間信号に対する離散フーリエ変換では，得られる周波数成分は $k = 0, 1, \ldots, N-1$ の**ちょうど** N **個**に限られ，有限和で表される。これは標本化により表現できる最高周波数が制限されること（標本化定理）に対応しており，N 個の標本値から N 個の周波数成分が得られるという意味で情報量が保存されている。