

情報工学

予想問題 第8回 解答

(グラフ・確率統計重視年)

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) (a) $d[u] + W[u][v] < d[v]$ (b) $d[u] + W[u][v]$

確定した頂点 u を経由することで頂点 v への距離が短くなるならば、その値で更新する（緩和, relaxation）。

(1-2) 始点 A （頂点 0）から実行する。選ばれる頂点は、未確定頂点のうち d が最小のものである。

反復	選択	$d[A]$	$d[B]$	$d[C]$	$d[D]$	$d[E]$	$d[F]$
初期	—	0	∞	∞	∞	∞	∞
1	A	0	4	2	∞	∞	∞
2	C	0	4	2	∞	5	∞
3	B	0	4	2	14	5	∞
4	E	0	4	2	9	5	∞
5	D	0	4	2	9	5	20
6	F	0	4	2	9	5	20

選ばれる順は

$$A \rightarrow C \rightarrow B \rightarrow E \rightarrow D \rightarrow F.$$

（反復 3 で B を確定した際に $d[D] = 4 + 10 = 14$ となるが、反復 4 で E を確定した際に $d[D] = 5 + 4 = 9$ へ更新される点に注意。）

(2-1) 最終的な d の値は頂点番号順に 0, 4, 2, 9, 5, 20 である。よって出力は以下のとおり。

0 4 2 9 5 20

(2-2) 条件：すべての辺の重みが非負であること。

ダイクストラ法は「現在未確定の頂点のうち距離最小のものは、それ以上短くならない」という性質に依拠している。辺の重みが非負であれば、他の頂点を經由しても距離が減ることはないためこの性質が成り立つ。

負の重みがある場合の反例：頂点 A, B, C と辺 $A \rightarrow B$ （重み 2）、 $A \rightarrow C$ （重み 5）、 $C \rightarrow B$ （重み -4 ）からなるグラフを考える。

始点 A から実行すると、まず $d[B] = 2$, $d[C] = 5$ となり、未確定のうち最小の B （距離 2）が確定する。しかし実際には $A \rightarrow C \rightarrow B$ の経路の長さが $5 + (-4) = 1$ であり、 $d[B]$ の真の最短距離は 1 である。 B を距離 2 で確定してしまったため、誤った結果となる。

(3-1) 時間計算量は $O(V^2)$ 。

根拠：26 行目のループは V 回繰り返される。各反復において、`minVertex` が全頂点を走査するため $O(V)$ 、さらに確定した頂点から出る辺の緩和のために全頂点を走査するため $O(V)$ を要する。したがって 1 反復あたり $O(V)$ であり、全体で $O(V) \times O(V) = O(V^2)$ となる。

(3-2) 2 分ヒープを用いた優先度付きキューでは、最小要素の取り出しが $O(\log V)$ 、距離の更新（キーの減少）が $O(\log V)$ で行える。取り出しは全体で V 回、更新は各辺につき高々 1 回で計 E 回生じる。よって

$$O((V + E) \log V).$$

（連結グラフでは $E \geq V - 1$ であるから $O(E \log V)$ と書いてもよい。）

(3-3) **疎なグラフ** ($E = O(V)$): ヒープを用いる実装は $O((V + V) \log V) = O(V \log V)$ となり, 配列走査による $O(V^2)$ より小さい。よって**ヒープを用いる実装が有利**である。

密なグラフ ($E = O(V^2)$): ヒープを用いる実装は $O(V^2 \log V)$ となり, $O(V^2)$ より $\log V$ 倍大きい。よって**配列走査による実装が有利**である。

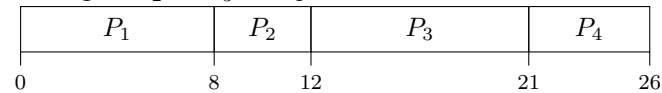
すなわち, 辺が少ないときはヒープ, 辺が多いときは単純な配列走査を選ぶとよい。

2 【必須問題】 計算機システムとシステムプログラム

(1-1) **実行状態から実行可能状態へ**：割り当てられた時間（タイムクォンタム）を使い切った場合、あるいはより優先度の高いプロセスが到着してプリエンプション（横取り）が起きた場合。CPU を手放すが、実行を続ける準備はできている。

実行状態から待ち状態へ：入出力の要求を発行した場合や、セマフォなどで資源の解放を待つ場合。事象の完了を待つ必要があり、CPU を与えられても実行を続けられない。

(1-2) **FCFS 方式**：到着順に $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4$ の順で実行する。



完了時刻は $P_1 : 8, P_2 : 12, P_3 : 21, P_4 : 26$ 。

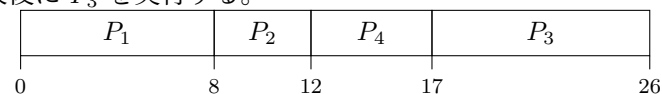
ターンアラウンド時間（完了時刻 - 到着時刻）は

$$P_1 : 8, P_2 : 11, P_3 : 19, P_4 : 23 \Rightarrow \text{平均} = \frac{8 + 11 + 19 + 23}{4} = 15.25 \text{ ms.}$$

待ち時間（ターンアラウンド時間 - 実行時間）は

$$P_1 : 0, P_2 : 7, P_3 : 10, P_4 : 18 \Rightarrow \text{平均} = \frac{0 + 7 + 10 + 18}{4} = 8.75 \text{ ms.}$$

SJF 方式（非プリエンプティブ）：時刻 0 では P_1 のみが到着しているので P_1 を実行する。時刻 8 の時点で P_2 (4), P_3 (9), P_4 (5) が到着済みであり、実行時間最短の P_2 を選ぶ。次に時刻 12 で P_4 (5) と P_3 (9) のうち P_4 を選び、最後に P_3 を実行する。



完了時刻は $P_1 : 8, P_2 : 12, P_4 : 17, P_3 : 26$ 。

ターンアラウンド時間は

$$P_1 : 8, P_2 : 11, P_3 : 24, P_4 : 14 \Rightarrow \text{平均} = \frac{8 + 11 + 24 + 14}{4} = 14.25 \text{ ms.}$$

待ち時間は

$$P_1 : 0, P_2 : 7, P_3 : 15, P_4 : 9 \Rightarrow \text{平均} = \frac{0 + 7 + 15 + 9}{4} = 7.75 \text{ ms.}$$

(1-3) 優れている理由：実行時間の短いプロセスを先に処理すると、それを待っているプロセスの数が早く減る。長いプロセスを先に実行すると後続のすべてのプロセスがその長時間を待つことになるため、短いものを優先する方が待ち時間の総和が小さくなる。（実際、非プリエンプティブな SJF は平均待ち時間を最小にすることが証明されている。）

適用が難しい理由：各プロセスの実行時間を事前に正確に知ることができないため。（実際には過去の実行履歴から指数平滑法などで推定する。）

(1-4) 現象の名称：**飢餓** (starvation, 無限延期)。

回避手法の例：**エイジング** (aging)。待ち状態にあるプロセスの優先度を、待ち時間の経過に応じて徐々に高くしていく。これにより、どれほど実行時間の長いプロセスであっても十分待てば最高優先度に達し、いずれ必ず実行される。

(2-1) 可変区画方式で生じる現象の名称：**外部断片化** (external fragmentation)。

固定区画方式で生じる無駄の名称：**内部断片化** (internal fragmentation)。

生じ方：固定区画方式ではあらかじめ決められた大きさの区画にプロセスを割り当てるため、プロセスの大きさが区画の大きさより小さい場合、その差分が使われないまま区画内に残る。この使えない領域が内部断片化である。

(2-2) 空き領域は先頭から 200, 80, 300, 120（単位は KiB）である。

最初適合方式：先頭から走査し、最初に見つかった十分な大きさの領域へ割り当てる。

要求	割り当て先	割り当て後の空き領域
90	第 1 領域 (200)	110, 80, 300, 120
180	第 3 領域 (300)	110, 80, 120, 120
100	第 1 領域 (110)	10, 80, 120, 120

すべての要求を満たせた。最終的な空き領域は 10, 80, 120, 120 である。

最良適合方式：要求を満たす領域のうち最も小さいものに割り当てる。

要求	割り当て先	割り当て後の空き領域
90	第 4 領域 (120, 最小)	200, 80, 300, 30
180	第 1 領域 (200, 最小)	20, 80, 300, 30
100	第 3 領域 (300)	20, 80, 200, 30

こちらもすべての要求を満たせた。最終的な空き領域は 20, 80, 200, 30 である。

両方式ともすべての要求を処理できたが、最良適合方式では 20, 30 といた非常に小さく再利用しにくい領域が生じている点が異なる。

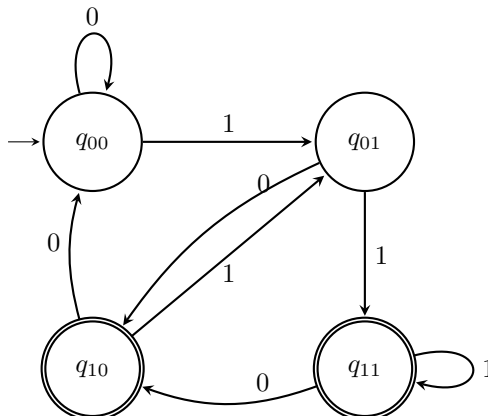
(2-3) **長所**：プログラムの論理的な構造（手続き、配列、スタックなど）に対応した単位で分割するため、共有や保護をその単位で行える。例えば手続きのセグメントを複数プロセスで共有したり、読み出し専用を設定したりすることが自然に行える。

短所：セグメントの大きさが可変であるため、主記憶上に連続領域を確保する必要があり、外部断片化が生じる。

組み合わせる理由：セグメンテーションの論理的な分割の利点を保ちながら、各セグメントを内部でページに分割してページング方式で管理すれば、連続領域を確保する必要がなくなり外部断片化を解消できる。すなわち両者の長所を同時に得られるためである。

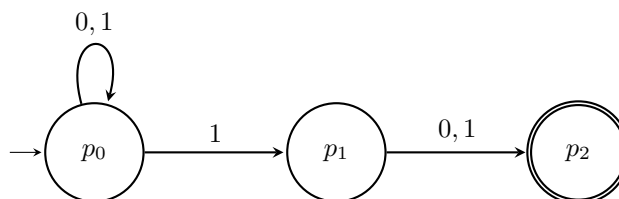
4 【選択問題】 計算理論

(1-1) L_2 は「末尾から 2 番目の文字が 1 である語」の集合である。DFA は直前 2 文字を記憶する必要があるため 4 状態となる。状態を (c_1, c_0) (c_1 が 2 つ前, c_0 が 1 つ前に読んだ文字) とし, $q_{00}, q_{01}, q_{10}, q_{11}$ と表す。開始状態は q_{00} , 受理状態は q_{10}, q_{11} (末尾から 2 番目, すなわち c_1 が 1) である。



(開始状態 q_{00} からは「まだ 2 文字読んでいない」場合も同じ扱いでよい。 $|w| < 2$ の語は受理されない。)

(1-2) NFA では「ここが末尾から 2 番目である」と非決定的に推測できるので 3 状態で済む。



差が生じる理由：NFA は p_0 で「いま読む 1 が末尾から 2 番目である」と推測して p_1 へ移り, あと 1 文字読んで入力が終われば受理する。すなわち推測が当たった計算だけを考えればよい。一方 DFA は推測ができないため, 直前 2 文字の組合せ $2^2 = 4$ 通りをすべて状態として保持しなければならない。

(2-1) $x \neq y$ かつ $|x| = |y| = k$ であるから, x と y が異なる位置が存在する。その最初の位置を先頭から数えて m 番目とする ($1 \leq m \leq k$)。一般性を失わず, x の m 文字目が 1, y の m 文字目が 0 であるとしてよい。

ここで $z = 0^{m-1}$ (長さ $m-1$) ととる。このとき

$$|xz| = |yz| = k + (m-1) = k + m - 1$$

である。 xz の末尾から k 番目の文字は, 先頭から数えて

$$|xz| - k + 1 = (k + m - 1) - k + 1 = m$$

番目の文字である。 $m \leq k$ であるからこの位置は x の内部にあり, すなわち x の m 文字目に一致する。同様に yz の末尾から k 番目の文字は y の m 文字目である。

x の m 文字目は 1 であるから $xz \in L_k$, y の m 文字目は 0 であるから $yz \notin L_k$ である。よって $z = 0^{m-1}$ が求める語であり, xz と yz の一方だけが L_k に属する。☒

(2-2) L_k を認識する DFA M が存在し, その状態数が 2^k 未満であると仮定する。

長さ k の語は全部で 2^k 個ある。開始状態からこれら 2^k 個の語を読んだときに到達する状態を考えると, 状態数が 2^k 未満であるから, 鳩の巣原理により, 相異なる 2 つの語 $x, y \in \Sigma^k$ ($x \neq y$) が存在して同じ状態 q に到達する。

小問 (2-1) より, ある語 z が存在して xz と yz の一方だけが L_k に属する。ところが x と y はともに状態 q に到達しているので, そこから同じ語 z を読めば必ず同じ状態に至る。したがって M は xz と yz をともに受理するか, ともに受理しないかのいずれかであり, 一方だけを受理することはできない。

これは $L(M) = L_k$ であることに矛盾する。よって仮定が誤りであり, L_k を認識する任意の DFA は少なくとも 2^k 個の状態をもつ。☒

(3-1) $L = \{0^i 1^j \mid i \neq j\}$ の補集合 $\bar{L}$ は, Σ^* のうち L に属さない語全体である。これには「 $0^i 1^j$ の形をしていない語」と「 $0^i 1^j$ の形で $i = j$ であるもの」の両方が含まれる。

$L(0^* 1^*)$ は「0 がすべて 1 より前に並んだ語」全体, すなわち $0^i 1^j$ の形の語全体である。したがって積をとると前者が除かれ

$$\bar{L} \cap L(0^* 1^*) = \{0^n 1^n \mid n \geq 0\}.$$

(3-2) L が正則言語であると仮定する。

正則言語は補集合について閉じているから $\bar{L}$ も正則言語である。また $0^* 1^*$ は正則表現で表されるので $L(0^* 1^*)$ は正則言語である。正則言語は積集合について閉じているから

$$\bar{L} \cap L(0^* 1^*)$$

も正則言語となる。

しかし小問 (3-1) よりこれは $\{0^n 1^n \mid n \geq 0\}$ であり, 正則言語でないことが知られている。矛盾。

したがって仮定が誤りであり, L は正則言語ではない。☒

6 【選択問題】電子回路と論理設計

(1-1) 非反転入力端子 (+) は接地されているので電位は 0 である。理想的な演算増幅器では仮想短絡により二つの入力端子の電位が等しくなるので、反転入力端子 (-) の電位も 0 となる。実際に配線で接地されているわけではないため、これを**仮想接地** (virtual ground) と呼ぶ。

(1-2) 反転入力端子の電位が 0 であるから、 R_1 を流れる電流は

$$i_1 = \frac{v_i - 0}{R_1} = \frac{v_i}{R_1}.$$

理想的な演算増幅器では入力端子に電流が流れ込まないので、この電流はすべて R_2 を通って出力側へ流れる。 R_2 の両端の電位は入力側が 0、出力側が v_o であるから

$$i_2 = \frac{0 - v_o}{R_2} = -\frac{v_o}{R_2}.$$

$i_1 = i_2$ より

$$\frac{v_i}{R_1} = -\frac{v_o}{R_2} \implies v_o = -\frac{R_2}{R_1} v_i.$$

回路の名称：**反転増幅回路** (inverting amplifier)。

(1-3) 電圧利得は

$$\frac{v_o}{v_i} = -\frac{R_2}{R_1} = -\frac{100 \text{ k}\Omega}{10 \text{ k}\Omega} = -10.$$

$v_i = 0.2 \text{ V}$ のとき

$$v_o = -10 \times 0.2 = -2.0 \text{ V}.$$

(1-4) R_2 を取り除くと帰還経路がなくなり、演算増幅器は開ループで動作する。開ループ利得は非常に大きい (10^5 程度以上) ので、入力端子間にわずかな電位差があっても出力はただちに電源電圧の限界まで振り切れる。すなわち出力は正または負の飽和状態となり、増幅器として機能しない。

負帰還の役割：出力の一部を反転入力へ戻すことで、二つの入力端子の電位差が 0 に保たれるように回路が自動的に動作する。その結果、利得が演算増幅器自体の特性（ばらつきが大きく温度にも依存する）ではなく、外付けの抵抗比 R_2/R_1 という安定した値のみで決まるようになる。

(2-1) 青が 3 サイクル、黄が 1 サイクル、赤が 2 サイクルであるから、必要な状態数は

$$3 + 1 + 2 = 6.$$

状態遷移図は、 $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_5 \rightarrow S_0$ と一巡する単純な環状である。各状態の出力は次のとおり。

状態	色	g	y	r
S_0	青	1	0	0
S_1	青	1	0	0
S_2	青	1	0	0
S_3	黄	0	1	0
S_4	赤	0	0	1
S_5	赤	0	0	1

(2-2) 状態割り当ては $S_0 = 000$, $S_1 = 001$, $S_2 = 010$, $S_3 = 011$, $S_4 = 100$, $S_5 = 101$ である。遷移表は次のとおり。

y_2	y_1	y_0	D_2	D_1	D_0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	ドントケア		
1	1	1	ドントケア		

カルノー図でドントケアを活用して簡単化すると

$$D_2 = y_1 y_0 + y_2 \overline{y_0}, \quad D_1 = \overline{y_2} (y_1 \oplus y_0), \quad D_0 = \overline{y_0} (\overline{y_2} + \overline{y_1}).$$

D_1 を積和形で書くと $D_1 = \overline{y_2} \overline{y_1} y_0 + \overline{y_2} y_1 \overline{y_0}$, D_0 を積和形で書くと $D_0 = \overline{y_2} \overline{y_0} + \overline{y_1} \overline{y_0}$ である。

(検証: $(0, 1, 1) \rightarrow D_2 = 1 \cdot 1 + 0 = 1$, $D_1 = 1 \cdot (1 \oplus 1) = 0$, $D_0 = 0$ となり $S_3 \rightarrow S_4 = 100$ と一致する。他の状態も同様に確認できる。)

(2-3) 出力は状態のみで定まる (Moore 型)。

$r = 1$ となるのは $S_4 = 100$ と $S_5 = 101$ であり, これは $y_2 = 1$ と同値である ($y_2 = 1$ の他の組合せ 110, 111 はドントケア)。よって

$$r = y_2.$$

$y = 1$ となるのは $S_3 = 011$ のみであるから

$$y = \overline{y_2} y_1 y_0.$$

$g = 1$ となるのは S_0, S_1, S_2 , すなわち $y_2 = 0$ かつ $(y_1, y_0) \neq (1, 1)$ のときであるから

$$g = \overline{y_2} \overline{(y_1 y_0)} = \overline{y_2} (\overline{y_1} + \overline{y_0}).$$

積和形では $g = \overline{y_2} \overline{y_1} + \overline{y_2} \overline{y_0}$ である。

7 【選択問題】 数学解析と信号処理

(1-1) 尤度関数は

$$L_f(\mu, \sigma^2) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right).$$

対数をとると、積が和になり

$$\begin{aligned} \log L_f(\mu, \sigma^2) &= \sum_{i=1}^n \left[-\frac{1}{2} \log(2\pi\sigma^2) - \frac{(x_i - \mu)^2}{2\sigma^2} \right] \\ &= -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2. \end{aligned}$$

(1-2) μ について： μ を含むのは第 2 項のみである。

$$\frac{\partial}{\partial \mu} \log L_f = -\frac{1}{2\sigma^2} \sum_{i=1}^n 2(x_i - \mu)(-1) = \frac{1}{\sigma^2} \sum_{i=1}^n (x_i - \mu).$$

これを 0 とおくと $\sum_i (x_i - \mu) = 0$ ，すなわち $\sum_i x_i = n\mu$ より

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n x_i.$$

 σ^2 について： $v = \sigma^2$ とおくと

$$\log L_f = -\frac{n}{2} \log(2\pi v) - \frac{1}{2v} \sum_{i=1}^n (x_i - \mu)^2$$

であるから

$$\frac{\partial}{\partial v} \log L_f = -\frac{n}{2v} + \frac{1}{2v^2} \sum_{i=1}^n (x_i - \mu)^2.$$

これを 0 とおくと

$$\frac{n}{2v} = \frac{1}{2v^2} \sum_i (x_i - \mu)^2 \implies v = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2.$$

 μ に $\hat{\mu}$ を代入して

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})^2.$$

(1-3) $\mathcal{D} = \{2, 4, 4, 4, 5, 5, 7, 9\}$ ， $n = 8$ である。

$$\hat{\mu} = \frac{2 + 4 + 4 + 4 + 5 + 5 + 7 + 9}{8} = \frac{40}{8} = 5.$$

偏差の 2 乗和は

$$(2 - 5)^2 + (4 - 5)^2 \times 3 + (5 - 5)^2 \times 2 + (7 - 5)^2 + (9 - 5)^2 = 9 + 3 + 0 + 4 + 16 = 32.$$

よって

$$\hat{\sigma}^2 = \frac{32}{8} = 4.$$

(標準偏差は $\hat{\sigma} = 2$ 。)

(1-4) 不偏分散は

$$s^2 = \frac{1}{n-1} \sum_i (x_i - \hat{\mu})^2 = \frac{32}{7} \approx 4.57.$$

$\hat{\sigma}^2 = 4$ と比べると s^2 の方が大きい。これは、最尤推定値 $\hat{\sigma}^2$ が標本平均 $\hat{\mu}$ を用いて偏差を測っているため、母平均を用いた場合より偏差の 2 乗和が小さくなり、分散を過小評価する（負の偏りをもつ）ことによる。

n が大きくなるとき： $\frac{s^2}{\hat{\sigma}^2} = \frac{n}{n-1} \rightarrow 1$ であるから、両者の差は 0 に近づく。すなわち標本数が多ければ偏りは実用上無視できる。

(2-1) 定義より

$$W(e^{j\omega}) = \sum_{n=0}^{N-1} e^{-j\omega n} = \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}}.$$

分子・分母をそれぞれ半角の指数でくくると

$$1 - e^{-j\omega N} = e^{-j\omega N/2} \cdot 2j \sin \frac{N\omega}{2}, \quad 1 - e^{-j\omega} = e^{-j\omega/2} \cdot 2j \sin \frac{\omega}{2}$$

であるから

$$W(e^{j\omega}) = e^{-j\omega(N-1)/2} \frac{\sin(N\omega/2)}{\sin(\omega/2)}.$$

(2-2) $|W(e^{j\omega})| = \left| \frac{\sin(N\omega/2)}{\sin(\omega/2)} \right|$ が 0 となるのは、分母が 0 でない範囲で $\sin(N\omega/2) = 0$ のときである。
 $0 < \omega \leq \pi$ では

$$\frac{N\omega}{2} = \pi \implies \omega = \frac{2\pi}{N}$$

が最初の零点である。

主ローブは $\omega = 0$ を中心とし、 $\omega = \pm 2\pi/N$ で 0 となる。したがって主ローブの幅は

$$\frac{2\pi}{N} - \left(-\frac{2\pi}{N}\right) = \frac{4\pi}{N}.$$

N を大きくすると主ローブの幅は $1/N$ に比例して狭くなる。

(2-3) 二つの正弦波の正規化角周波数の差を $\Delta\omega$ とする。有限長で観測すると、各正弦波のスペクトルは線スペクトルではなく主ローブの幅をもつ山として現れる。二つの山を分離して観測するには、主ローブの幅が周波数差より小さい必要がある。すなわち

$$\frac{4\pi}{N} < \Delta\omega \iff N > \frac{4\pi}{\Delta\omega}$$

となるように観測長 N を十分大きくとる必要がある。周波数差が小さいほど長い観測が必要となる。

副ローブの悪影響：矩形窓の副ローブは主ローブに比べて十分小さいとはいえず（第 1 副ローブは主ローブの約 -13 dB）、強い信号の副ローブが近傍の周波数に漏れ出す。この漏れが弱い信号のスペクトルを覆い隠してしまい、振幅の大きく異なる二つの信号が共存する場合に弱い方を検出できなくなる。（これを軽減するにはハミング窓など副ローブの小さい窓関数を用いる。）