専門科目（午前）　　　　　　　　　　　　２７　大修

情報工学　　　　　　　　　　　　　　時間　9：30 ～ 12：30

通信情報工学専攻・計算工学専攻

注　意　事　項

1. 各解答用紙に受験番号を記入せよ.
2. 次の５題をすべて解答せよ.
3. 解答は１題ごとに別々の解答用紙に，問題番号を明記した上で記入せよ．必要であれば，解答用紙の裏面に記入してよいが，解答用紙の表面にその旨を明記すること.
4. １枚の解答用紙に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枚以上の解答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\. 1) $\boldsymbol{T} = \begin{pmatrix} 0.5 & 0.4 & 0.3 \\ 0.4 & 0.5 & 0.3 \\ 0.1 & 0.1 & 0.4 \end{pmatrix}$, $\boldsymbol{A} = \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{pmatrix}$, $\boldsymbol{b} = \begin{pmatrix} 0.3 \\ 0.3 \end{pmatrix}$ に対して，以下の問に答えよ．

a) $\boldsymbol{T}^2$ を求めよ．

b) $\boldsymbol{A}$ のすべての固有値と，それに対応する固有ベクトルを求めよ．

c) 正の整数 N に対して，$\boldsymbol{A}^N\boldsymbol{b}$ および $\sum_{n=0}^{N} \boldsymbol{A}^n\boldsymbol{b}$ を求めよ．

2) 図 1.1 に示したグラフのノード W_A，W_B，W_C はAさん，Bさん，Cさんのウェブページを表し，エッジはハイパーリンク，エッジに付随する数値はそのハイパーリンクをクリックする確率を表す．

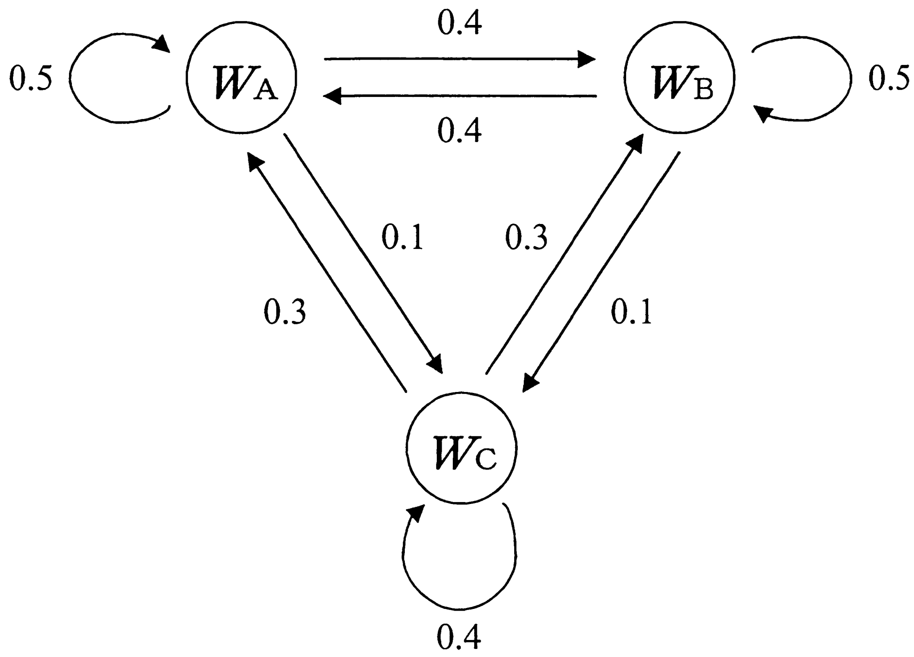

図 1.1

a) W_A からハイパーリンクを 2 回クリックした後に再び W_A にいる確率を求めよ．

b) ハイパーリンクを N 回（$N \geq 1$）クリックした後に W_A および W_B にいる確率を，P_N および Q_N とする．P_{N-1} と Q_{N-1} を用いて P_N および Q_N を表せ．

c) 最初に W_A にいる確率 P_0 および最初に W_B にいる確率 Q_0 がともに 0.3 のとき，P_N および Q_N を N の関数として求めよ．

d) $\lim_{N\to\infty} P_N$ および $\lim_{N\to\infty} Q_N$ を求めよ．

2. 関数 $f(x) = \mathrm{sgn}(x) + \cos \pi x + \sin 5\pi x \quad (-1 \leq x < 1)$ について以下の問に答えよ．
ただし，$\mathrm{sgn}(x)$ は

$$\mathrm{sgn}(x) = \begin{cases} -1 & (-1 \leq x < 0) \\ 1 & (0 \leq x < 1) \end{cases}$$

である．

1) ２つの実数 α と β を用いて関数 $g(x) = \alpha \cos \pi x + \beta \sin 5\pi x$ を構成するとき，

$$E(\alpha, \beta) = \int_{-1}^{1} |f(x) - g(x)|^2 \, dx$$

を α と β の２次関数として表せ．

2) $E(\alpha, \beta)$ が最小となるときの α と β と $E(\alpha, \beta)$ の値を各々求めよ．

3) 上記の関数 $f(x)$ を用いて周期関数 $\tilde{f}(t)$ $(-\infty < t < \infty)$ を $\tilde{f}(x + 2m) = f(x)$ $(-1 \leq x < 1$, m は任意の整数) のように定義する．$\tilde{f}(t)$ を $\frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n \cos n\pi t + b_n \sin n\pi t)$ の形にフーリエ級数展開するとき，フーリエ係数 a_n $(n = 0, 1, 2, \ldots)$ と b_n $(n = 1, 2, \ldots)$ を各々求めよ．

4) 3) で求めた周期関数 $\tilde{f}(t)$ のフーリエ級数に現れる全ての項に $t = \frac{1}{2}$ と $t = 55$ と $t = 100$ を代入した場合の極限値

$$\begin{aligned} A &= \frac{a_0}{2} + \lim_{N \to \infty} \sum_{n=1}^{N} \left(a_n \cos \frac{1}{2} n\pi + b_n \sin \frac{1}{2} n\pi \right) \\ B &= \frac{a_0}{2} + \lim_{N \to \infty} \sum_{n=1}^{N} (a_n \cos 55 n\pi + b_n \sin 55 n\pi) \\ C &= \frac{a_0}{2} + \lim_{N \to \infty} \sum_{n=1}^{N} (a_n \cos 100 n\pi + b_n \sin 100 n\pi) \end{aligned}$$

を各々理由を付して求めよ．必要であれば,「フーリエ級数の各点収束定理」の結果を証明せずに用いてよい．

3. 時刻 t の実関数である受信信号 $r(t)$ をサンプリングし，送信シンボル b_0 を判定する問題を考える．まず，$r(t)$ を

$$r(t) = b_0 a(t) + n(t) \tag{3.1}$$

と定める．ただし，b_0 は ± 1 の 2 値をとる確率変数とし，$a(t)$ のフーリエ変換 $A(f) = \int_{-\infty}^{\infty} a(t) e^{-j2\pi ft} dt$ を

$$A(f) = \begin{cases} \frac{1}{2f_M} & |f| \le f_M \\ 0 & |f| > f_M \end{cases} \tag{3.2}$$

とする．ここで，j は虚数単位，f_M は $A(f)$ の最高周波数で正の定数である．また，$n(t)$ は雑音であり，時刻 t によらず平均 0，分散 σ^2 のガウス分布に従うものとする．すなわち，$n(t)$ の確率密度関数 $p[n(t)]$ は

$$p[n(t)] = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{n(t)^2}{2\sigma^2}\right] \tag{3.3}$$

となる．なお，$n(t)$ と b_0 は統計的に独立と仮定する．以下の問に答えよ．

1) $a(t) = \int_{-\infty}^{\infty} A(f) e^{j2\pi ft} df$ である．x が実数のとき，$\sin x = \frac{e^{jx} - e^{-jx}}{2j}$ となることを用いて，$a(t)$ を j を含まない形で求めよ．

2) $|a(t)|$ が最小となる時刻 t の条件を求めよ．

3) $|a(t)|$ が最大となる時刻 t を 1 つ求めよ．

4) $r(t)$ を用いて b_0 の信号判定を行う．ただし，$a(t) \neq 0$ を満たす時刻 t に限る．$a(t)r(t) > 0$ のとき $b_0 = 1$，$a(t)r(t) \le 0$ のとき $b_0 = -1$ と判定する．$b_0 = 1$ の場合，b_0 の判定誤りが起きる確率 P_{1e} は

$$P_{1e} = \frac{1}{\sqrt{\pi}} \int_{\sqrt{\gamma}}^{\infty} e^{-x^2} dx \tag{3.4}$$

となることを示せ．ただし，$\gamma = \frac{a(t)^2}{2\sigma^2}$ である．

同様に，$b_0 = -1$ の場合，b_0 の判定誤りが起きる確率 P_{-1e} を求めよ．

5) $b_0 = 1$ となる確率を q $(0 \le q \le 1)$ とする．$a(t) \neq 0$ を満たす時刻 t において，b_0 の判定誤りが起きる確率 P_e を求めよ．

6) $a(t) \neq 0$ を満たす時刻 t のうち，P_e を最小にする t を求めよ．また，その理由についても説明せよ．

4.

1) 以下の論理代数（ブール代数）の等式を証明せよ．ただし $x \vee y$ は x と y の論理和（OR），xy は x と y の論理積（AND），$\overline{x}$ は x の否定（NOT）を表す．

a) $(x_1 \vee \overline{x_2})(x_2 \vee \overline{x_3})(x_3 \vee \overline{x_1}) = x_1x_2x_3 \vee \overline{x_1}\,\overline{x_2}\,\overline{x_3}$

b) $(x_1 \vee \overline{x_2})(x_2 \vee \overline{x_3})\cdots(x_{n-1} \vee \overline{x_n})(x_n \vee \overline{x_1}) = x_1x_2\cdots x_n \vee \overline{x_1}\,\overline{x_2}\cdots\overline{x_n} \qquad (n \geq 2)$

2) 4 進同期式カウンタを実現する順序回路の設計を考える．入力を x，出力を y の各 1 ビット，現在の状態を $q_1\,q_0$，次の状態を $q_1'\,q_0'$ の各 2 ビットとする．初期状態は $q_1\,q_0 = 00$ とする．$x = 1$ が入力されるたびに，$q_1\,q_0$ の値は $00 \to 01 \to 10 \to 11 \to 00 \to \cdots$（以後，同じ繰り返し）と変化する．$x = 0$ の時，$q_1\,q_0$ の値は変化しない．$x = 1$ かつ $q_1\,q_0 = 11$ の時のみ $y = 1$ となり，それ以外では $y = 0$ とする．

表 4.1：カルノー図

x \ $q_1\,q_0$	00	01	11	10
0				
1				

a) q_1', q_0', y それぞれについて，x, q_1, q_0 によるカルノー図を示せ．カルノー図の作成には表 4.1 の形式を参考にせよ．

b) a) の解を用いて，x, q_1, q_0 に関する NOT-AND-OR 形式（「積項の和」形式，積和標準形）で q_1', q_0', y それぞれを表せ．ただし，項数が最小となるように簡単化すること．

c) b) の解を用いて，順序回路図を示せ．2 入力の NAND ゲート，3 入力の NAND ゲート，および立ち上がり動作のエッジトリガ型 D フリップフロップのみを用いること．クロックへの入力は順序回路図では省略して良い．

3) メモリキャッシュについて，空欄 $\boxed{\text{A}}$ 〜 $\boxed{\text{P}}$ に最も適切な語句を，図 4.3 の選択肢群から選んで，その番号を解答せよ．同じ選択肢を何度選んでも良い．

a) メモリキャッシュは CPU とメモリの間に配置する記憶装置であり，メモリよりも小容量で高速に動作する．一般的にメモリアクセスには参照の $\boxed{\text{A}}$ があるため，メモリキャッシュにメモリ中のデータの一部を一時的に保存することで，全体の平均アクセス時間を向上させることができる．アクセスされたメモリがすぐに再度アクセスされる可能性が高いことを $\boxed{\text{B}}\boxed{\text{A}}$ といい，アクセスされたメモリと近い番地のメモリが，同時あるいはすぐにアクセスされる可能性が高いことを $\boxed{\text{C}}\boxed{\text{A}}$ という．

b) メモリキャッシュの書き込み操作には大きく 2 つの方式がある．$\boxed{\text{D}}$ 方式は書き込み操作をメモリキャッシュとメモリの両方に同時に行うため，メモリキャッシュとメモリの内容が常に一貫しているが，書き込み速度は向上しない．一方，$\boxed{\text{E}}$ 方式はメモリキャッシュのみに書き込みを行い，メモリへの書き込みはフラッシュ時などに遅延して実行されるため，書き込み速度が向上する．

（次ページにつづく）

（前ページのつづき）

c) メモリ中のデータをメモリキャッシュのどこに格納することを許すか，すなわちメモリとメモリキャッシュの対応づけの違いで，大きく3つの方式がある． F 方式はあるメモリのデータの格納をメモリキャッシュの一箇所のみに許す（図4.2の①）． G 方式はメモリキャッシュ内で事前に限定された一部の複数箇所への格納を許し，その複数箇所から1つを選んで格納する（図4.2の②）． H 方式は全ての場所への格納を許し，1つを選んで格納する（図4.2の③）．この3つの方式のうち，一般的に最もキャッシュミスしやすいのは I 方式である．またキャッシュミスしない場合のアクセス時間が最も短いのは J 方式である．

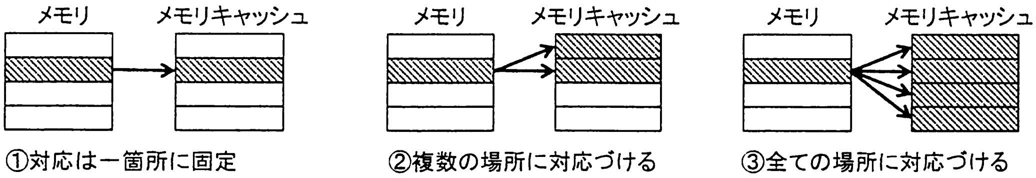

図4.2：メモリとメモリキャッシュの対応関係

d) キャッシュ置換アルゴリズムとは，メモリ中のデータを新たにメモリキャッシュに格納するために，メモリキャッシュ中から捨てるデータを選択するアルゴリズムのことであり，様々な方式がある． K は最近のある一定期間内のアクセス頻度が最少のデータを捨てる． L はメモリキャッシュにデータを格納した時刻が最も古いデータを捨てる． M は最後のアクセス時刻が最も古いデータを捨てる．

e) メモリ中のデータをメモリキャッシュに転送する方式には大きく2つの方式がある． N はプログラムがデータにアクセスしたときにそのデータを転送する．一方， O はプログラムがデータにアクセスする前にそのデータを転送する．この2つの方式のうち， B A が無く，次にアクセスするデータを予測しやすい処理の場合，例えば巨大な配列の各要素を先頭から逐次的に一回ずつアクセスする場合などは， P の方が性能の向上を期待できる．

(1) インダイレクト・マップ，(2) 完全性，(3) 局所性，(4) 空間的，(5) 健全性，(6) 構造的，(7) 時間的，(8) スワッピング，(9) 脆弱性，(10) セグメンテーション，(11) 絶対的，(12) セット・アソシアティブ，(13) 相対的，(14) ダイレクト・マップ，(15) デマンドフェッチ，(16) 透過性，(17) バッファリング，(18) ビジーウェイト，(19) 物理的，(20) プリフェッチ，(21) フル・アソシアティブ，(22) ページング，(23) ポーリング，(24) ライトスルー，(25) ライトバック，(26) ラウンドロビン，(27) ランダム，(28) 論理的，(29) 割り込み，(30) FIFO，(31) LFU，(32) LRU

図4.3：選択肢群

5. a を整数列とし，その項を a_i（i は非負整数）で表す．n を $n \geq 1$ をみたす整数とする．a の最初の n 項 $a_0, a_1, \ldots, a_{n-1}$ における，長さ 1 以上の連続する項の和で最大のものを $f(a,n)$ とすると，これは以下のように定義できる．

$$f(a,n) = \max\left\{ \sum_{k=i}^{j} a_k \;\middle|\; 0 \leq i \leq j < n \right\}.$$

例えば $a = (20, -24, 36, 3, -7, 27, -40, 37, -28, 6)$ の場合，$f(a,10) = 36 + 3 + (-7) + 27 = 59$ となる．以下，$f(a,n)$ を計算する関数のプログラミング言語 C による定義を試みる．整数列 a は配列 a として与えられるものとする．与えられる配列は十分長く，範囲外の参照は生じないと仮定する．加えて，整数の演算についてオーバーフロー等は生じないと仮定する．また，引数 n の値は 1 以上であるとする．

1) まずは上の定義に従い，$f(a,n)$ を計算する関数 f および付随する関数を図 5.1 のように定義した．

```
int max(int x, int y) {
  return x > y ? x : y;
}

int sum(int a[], int i, int j) {
  int s = 0;
  for (int k = i; k <= j; k++)
    s = s + a[k];
  return s;
}
```

```
int f(int a[], int n) {
  int s = a[0];
  for (int i = 0; i < n; i++)
    for (int j = [ A ]; j < n; j++)
      s = max(s, sum(a, i, j));
  return s;
}
```

図5.1

a) 空欄 $\boxed{\text{A}}$ に入る式を答えよ．

b) 図 5.1 の関数 f による $f(a,n)$ の計算時間を $T_1(n)$ としたとき，$T_1(n) = O(t_1(n))$ をみたすとする．$t_1(n)$ にあてはまる式で n に対する増え方が最も小さいものを以下の選択肢から選べ．

$1, \quad \log n, \quad \sqrt{n}, \quad n, \quad n\log n, \quad n^2, \quad n^2\log n, \quad n^{\frac{5}{2}}, \quad n^3, \quad n^4, \quad 2^n, \quad 2^{2^n}. \qquad (*)$

ここで $T_1(n)$ は配列 a の要素を参照する回数 $A_1(n)$ および定数 c_1, c_1' を使って $T_1(n) = c_1 A_1(n) + c_1'$ と表せるものとする．なお，以下の問題においても関数の計算時間は同様に表せるものとする．

2) 図 5.1 のプログラムではあまりに計算に時間がかかると考えたので，少し工夫をして関数 f の定義を図 5.2 のように変更した．空欄 $\boxed{\text{A}}$ に入る式は問 1a) のものと同じとする．

```
int f(int a[], int n) {
  int s = a[0];
  for (int i = 0; i < n; i++) {
    int t = 0;
    for (int j = [ A ]; j < n; j++) {
      t = t + [ B ];
      s = max(s, t);
    }
  }
  return s;
}
```

図5.2

a) 空欄 $\boxed{\text{B}}$ に入る式を答えよ．

b) 図 5.2 の関数 f による $f(a,n)$ の計算時間を $T_2(n)$ としたとき，$T_2(n) = O(t_2(n))$ をみたすとする．$t_2(n)$ にあてはまる式で n に対する増え方が最も小さいものを問 1b) の選択肢 $(*)$ から選べ．

(次ページにつづく)

(前ページのつづき)

c) 図 5.1 から図 5.2 への変更によって計算時間のオーダーが小さくなる理由を 100 字程度までで述べよ.

3) 図 5.2 のプログラムはこれ以上工夫の余地がなさそうなので，ここで少し考え方を変えてみる．$n \geq 2$ である場合について，a の最初の n 項 $a_0, a_1, \ldots, a_{n-1}$ における連続した項の和を，末尾の項 a_{n-1} を含まない場合と含む場合に分けて考える．前者の最大値は $f(a, n-1)$ で与えられる．後者の最大値を $g(a,n)$ とすると，これは以下のように定義できる．

$$g(a,n) = \max\left\{ \sum_{k=i}^{n-1} a_k \,\middle|\, 0 \leq i < n \right\}.$$

$f(a,n)$ の定義より，$n \geq 2$ をみたす任意の n について $f(a,n) = \max\{f(a,n-1), g(a,n)\}$ がなりたつ．この考え方をもとに，$g(a,n)$ および $f(a,n)$ を計算する関数 g および f を図 5.3 のように定義した．

```
int g(int a[], int n) {
  int t = [ C ];
  for (int k = 1; k < n; k++)
    t = max(t + [ D ], [ D ]);
  return t;
}
```

```
int f(int a[], int n) {
  int s = a[0];
  for (int k = 1; k < n; k++)
    s = max([ E ], g(a, [ F ]));
  return s;
}
```

図5.3

a) 空欄 $\boxed{\mathrm{C}}$, $\boxed{\mathrm{D}}$, $\boxed{\mathrm{E}}$ および $\boxed{\mathrm{F}}$ に入る式を答えよ.

b) 図 5.3 の関数 f による $f(a,n)$ の計算時間を $T_3(n)$ としたとき，$T_3(n) = O(t_3(n))$ をみたすとする．$t_3(n)$ にあてはまる式で n に対する増え方が最も小さいものを問 1b) の選択肢 (*) から選べ.

4) 図 5.3 の 2 つの関数 g, f はよく似ている．そこでこれらを組み合わせ，$g(a,n)$ と $f(a,n)$ を一緒に計算することで計算時間を減らせると考えた．このアイデアにもとづいて定義した関数 f を図 5.4 に示す．空欄 $\boxed{\mathrm{C}}$, $\boxed{\mathrm{D}}$ および $\boxed{\mathrm{E}}$ に入る式は問 3a) のものと同じとする．

```
int f(int a[], int n) {
  int t = [ C ], s = a[0];
  for (int k = 1; k < n; i++) {
    t = max(t + [ D ], [ D ]);
    s = max([ E ], [ G ]);
  }
  return s;
}
```

図5.4

a) 空欄 $\boxed{\mathrm{G}}$ に入る式を答えよ.

b) 図 5.4 の関数 f による $f(a,n)$ の計算時間を $T_4(n)$ としたとき，$T_4(n) = O(t_4(n))$ をみたすとする．$t_4(n)$ にあてはまる式で n に対する増え方が最も小さいものを問 1b) の選択肢 (*) から選べ.

c) 図 5.3 から図 5.4 への変更によって計算時間のオーダーが小さくなる理由を 100 字程度までで述べよ.