平成 19 年 8 月 21 日（火）
13:00～16:00 実施

平成２０年度大学院前期課程入学試験

情報通信工学
入試問題

注意事項

・問題は７問、問題用紙の枚数は９枚（表紙を含まず）である。
・問題１～問題３は必須問題である。すべて解答すること。
・問題４～問題７は選択問題である。この中から２題を選択し、解答せよ。
・解答用紙は、

問題 1 を 1 枚目（白色）の解答用紙
問題 2 を 2 枚目（赤色）の解答用紙
問題 3 を 3 枚目（青色）の解答用紙
問題 4 を 4 枚目（黄色）の解答用紙
問題 5 を 5 枚目（水色）の解答用紙
問題 6 を 6 枚目（桃色）の解答用紙
問題 7 を 7 枚目（緑色）の解答用紙

に記入すること。選択問題については、選択した問題番号に対応する解答用紙を選択すること。解答用紙を間違えると採点されない場合があるので注意すること。

・試験終了後の解答用紙回収について

1) 試験終了後、指示に従い、選択しなかった問題の解答用紙を 2 つ折りにすること。
2) 選択問題については、すべてについて解答した後、2 つ折りにする段階で、選択問題を確定させても良い。
3) 解答した解答用紙、および 2 つ折りにした解答用紙は、試験監督者の指示に従って提出すること。提出する解答用紙は５枚、２つ折りにする解答用紙は２枚である。

情報理論

1－1 4ビットからなる情報ビット系列 $(x_1x_2x_3x_4)$ に対して，$c_1 = x_1 \oplus x_2 \oplus x_3$，$c_2 = x_2 \oplus x_3 \oplus x_4$，$c_3 = x_1 \oplus x_2 \oplus x_4$ となる3ビットの検査ビット $(c_1c_2c_3)$ を付加し，7ビットからなる符号語 $(x_1x_2x_3x_4c_1c_2c_3)$ を生成する符号化に関する以下の問いに答えよ．但し，上式中の $\oplus$ は排他的論理和演算を表す．

(i) 生成行列を求めよ．

(ii) 検査行列を求めよ．

(iii) 符号語 (1001000) が受信されたとする．この受信された符号語に誤りが含まれているかどうか答えよ．また，誤りが含まれている場合には訂正し，送信されたと判断される情報ビット系列を求めよ．

1－2 図1-1に示す通信路線図を有する二元対称通信路を通信路1とする．入力情報源記号は $A = \{0,1\}$，出力情報源記号は $B = \{0,1\}$ であり，通信路の遷移確率は図中に示す通りである (但し，$0 \leq p \leq 1$)．通信路1を2つ縦続に接続した通信路2(図1-2) に関する以下の問いに答えよ．

(i) 通信路2の通信路行列を求めよ．

(ii) 通信路2の通信路容量を求めよ (通信路2が二重に一様な通信路であることを利用して求めても良い)．

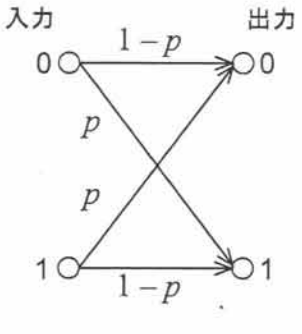


図1-1 通信路1

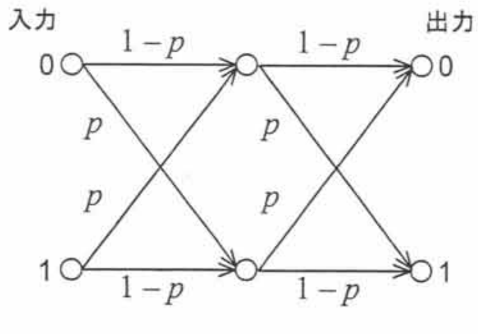


図1-2 通信路2

通信方式

2－1　確率密度関数及び累積分布について以下の問いに答えよ．

(i) 実数確率変数 X の確率密度関数 $p(x)$ が次式で与えられるとき，平均，分散および累積分布 $P(x)$ を求めよ．

$$p(x)=\begin{cases}1;\ 3\le x\le 4\\ 0;\ \text{その他}\end{cases}$$

(ii) 実数確率変数 X の確率密度関数が

$$p(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-m)^2}{2\sigma^2}}$$

で与えられるとき，その累積分布 $P(x)$ を導出せよ．ただし

$$\mathrm{erfc}(x)=\frac{2}{\sqrt{\pi}}\int_x^{\infty}e^{-t^2}dt,\ \mathrm{erfc}(\infty)=0,\ \mathrm{erfc}(0)=1$$

を利用せよ．

2－2　二進オンオフ・キーイング（Binary On-Off Keying）で変調された信号を受信機で受信する場合，雑音がない場合の受信波形は，送信データが「１」の場合，$A\cos(2\pi f_c t)$，送信データが「０」の場合，0 で表されるものとする．ただし，A は受信信号振幅，f_c は搬送周波数，t は時刻である．それに対して，受信時に発生する雑音は，

$$n(t)=x(t)\cos(2\pi f_c t)-y(t)\sin(2\pi f_c t)$$

で与えられ，$x(t)$，$y(t)$ は，いずれも，平均が 0，分散が σ^2 のガウス分布に従う，互いに独立な確率過程に従うものとする．この雑音が受信信号に加わった信号を同期検波するとき，以下の問いに答えよ．

(i) 雑音を含む受信信号 $z(t)$ を，送信データが「１」の場合と送信データが「０」の場合に分けて記せ．

(ii) 同期検波後の波形 $r(t)$ は，送信データが「１」の場合と送信データが「０」の場合とで，それぞれどのように表されるかを，その導出を含めて説明せよ．

(iii) 送信データが「１」の場合の $r(t)$ の条件付確率密度関数 $p(r\mid \text{データ}=1)$ と，送信データが「０」の場合の $r(t)$ の条件付確率密度関数 $p(r\mid \text{データ}=0)$ が数式でどのように表されるかを，その理由を含めて説明せよ．

(iv) 送信データにおける「１」と「０」の生起確率は等しいものとするとき，受信データの誤り率を求めよ． ただし，

$$\mathrm{erfc}(x)=\frac{2}{\sqrt{\pi}}\int_x^{\infty}e^{-t^2}dt,\ \mathrm{erfc}(\infty)=0,\ \mathrm{erfc}(0)=1$$

を利用せよ．また，受信機における「１」と「０」の判定は，最適なしきい値に対してなされるものとする．

ネットワーク工学

3．　M 台の送信ホストと 1 台の受信ホストが伝送路を共有し，送信ホストが以下に述べるスロット化アロハ（Slotted ALOHA）伝送方式に従いパケットを受信ホストに送出するシステムを考える．全送信ホストは時刻同期がとれているものと仮定する．伝送路は等時間間隔 τ のスロットに区切られ，送信ホストはスロット開始時刻においてパケットを送出するか否かを決定する．パケットを送出すると決定した場合には，そのスロット内で 1 つのパケットが送出されるものとする．各送信ホストから受信ホストまでの伝搬遅延時間は τ と比較して無視できる程度に小さいと仮定する．

各スロット内で 1 台の送信ホストからパケットが送出された場合，そのパケットは誤りなく受信ホストにおいて受信されるものとする．また，2 台以上の送信ホストからパケットが同一スロット内に送出された場合，パケット同士の衝突により，いずれのパケットも受信ホストにおいて受信されないものとする．このとき，送信ホストは自身が送出したパケットが衝突したことを衝突が発生したスロット内に知ることができるものと仮定する．

各送信ホストは，送出すべきパケットを無限個を持っていると仮定し，スロットの開始時刻に 2 つの状態 s_1，s_2 のいずれの状態にあるかで以下の手続きに従いパケットを送出するか否かを決定する．状態 s_1 は新規パケットを送出する状態であり，確率 p_1 $(0 < p_1 \leq 1)$ でパケットを送出する．送出したパケットが衝突した場合，そのホストの次のスロット開始時刻における状態は状態 s_2 となる．状態 s_2 は，衝突したパケットを再送する状態であり，確率 p_2 $(0 < p_2 < 1)$ で再送パケットを送出する．再送パケットが衝突せずに受信ホストにおいて受信された場合，そのホストの次のスロット開始時刻における状態は状態 s_1 となる．一方，再送パケットが衝突した場合，そのホストの次のスロット開始時刻における状態は状態 s_2 にとどまる．

あるスロットの開始時刻において，状態 s_1 であるホスト数が k，状態 s_2 であるホスト数が $M-k$ である場合，このスロットにおいてシステムは状態 k にあると呼ぶことにする．定常状態において，システムが状態 k にある確率を π_k $(k = 0, 1, \ldots, M)$ とすると，π_k は以下の式で与えられる．

$$\boldsymbol{\pi} = \boldsymbol{\pi P}$$

ただし，$\boldsymbol{\pi} = (\pi_0\ \pi_1\ \cdots\ \pi_M)$ であり，$\sum_{k=0}^{M} \pi_k = 1$ を満たす．また，$\boldsymbol{P}$ は $(M+1)$ 次正方行列であり，$\boldsymbol{P}$ の i 行 j 列成分 $P_{i,j}$ $(0 \leq i, j \leq M)$ は，状態 i から状態 j への遷移確率を表している．

以下の問いに答えよ．

(i) $\boldsymbol{P}$ の対角成分 $P_{i,i}$ $(0 \leq i \leq M)$ を i，M，p_1，p_2 を用いて示せ．また，$\boldsymbol{P}$ の M 行 $(M-1)$ 列成分 $P_{M,M-1}$ を示せ．

(ii) $\boldsymbol{P}$ の成分のうち，任意の p_1，p_2 $(0 < p_1 \leq 1, 0 < p_2 < 1)$ に対して 0 となる成分はいくつ存在するか，M を用いて示せ．

(iii) システムが状態 k $(k = 0, 1, \ldots, M)$ にあるという条件下で，パケットが送出され，かつそのパケットが衝突することなく受信される確率を P_k とする．P_k を k，p_1，p_2，M を用いて示せ．

(iv) スループットを定常状態における 1 スロット当たりの平均送信成功パケット数と定義する．$p_1 = p_2 = p$ である場合に，スループット F を最大とする p の値を求めよ．

(v) $p_1 = p_2 = p$ とする．このとき，$G = Mp$ を定数とし，M を限りなく大きくした場合のスループット F の極限値を示せ．ただし，

$$\lim_{x \to \infty} \left(1 + \frac{\alpha}{x}\right)^x = e^\alpha$$

を用いてもよい．

光・電波工学

4．電荷及び電流を含まない真空中におけるマクスウェルの方程式は，次のように表される．以下の問いに答えよ．

$$\nabla\times\boldsymbol{E}=-\mu_0\frac{\partial\boldsymbol{H}}{\partial t},\quad \nabla\times\boldsymbol{H}=\varepsilon_0\frac{\partial\boldsymbol{E}}{\partial t},\quad \nabla\cdot\varepsilon_0\boldsymbol{E}=0,\quad \nabla\cdot\mu_0\boldsymbol{H}=0$$

ここで $\boldsymbol{E}$ および $\boldsymbol{H}$ は，電界ベクトル及び磁界ベクトルを表し，ε_0，μ_0 は，真空の誘電率および透磁率を示す．また，$\nabla\times$ はベクトルの回転を求める演算を表し，$\nabla\cdot$ はベクトルの発散を求める演算を表す．

(i) 上のマクスウェルの方程式から，電界 $\boldsymbol{E}$ を定めるベクトル波動方程式を導け．

(ii) 本問ならびに以下のすべての問いで，直角座標系を用いるものとし，電磁界は xy 面内で一様とする．電界が x 方向成分のみをもつとき，電界の x 方向成分を定める一次元の波動方程式を導け．

(iii) $E_x(z,t)=E_+(t-\frac{z}{v})+E_-(t+\frac{z}{v})$ （ここで，E_+ 及び E_- は，任意の関数である）で与えられる電界の x 方向成分 $E_x(z,t)$ が，(ii)の一次元の波動方程式の解となることを示せ．ただし，$v=\dfrac{1}{\sqrt{\varepsilon_0\mu_0}}$ とする．

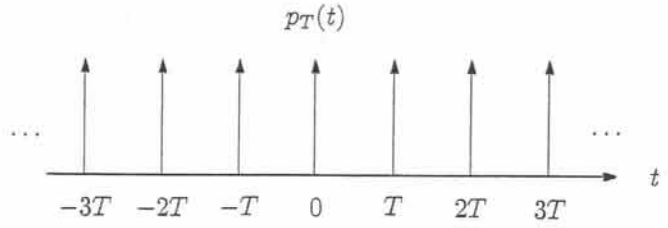
(iv) 一次元の波動方程式の解 $E_x(z,t)=E_+(t-\frac{z}{v})+E_-(t+\frac{z}{v})$ における E_+ 及び E_- が，それぞれ z の正および負の方向に速度 v で伝搬する波動を表すことを示せ．

(v) $E_x(z,t)=E_+(t-\frac{z}{v})$ のとき，$\sqrt{\dfrac{\mu_0}{\varepsilon_0}}H_y(z,t)=E_x(z,t)$ であることを示せ．ただし，

H_y は磁界の y 方向成分とする．

信号処理

5．連続時間信号 $x(t)$ のサンプリングを考える．無限に続く周期 T のサンプリングパルス列信号 $p_T(t)$ （サンプリング周波数 $f_S = \dfrac{1}{T}$）を下図のようにする．以下の問いに答えよ．



(i) 単位インパルス信号（ディラックのデルタ関数 $\delta(t)$）を用いて $p_T(t)$ を表し，フーリエ級数展開せよ．

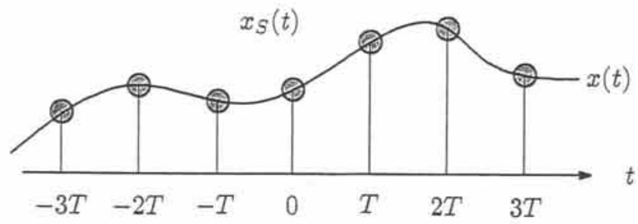
ここで，$x(t)$ のフーリエ変換を $X(\omega)$ とし，

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

と定義する（j は虚数単位）．フーリエ変換対を $x(t) \leftrightarrow X(\omega)$ と書く．また $X(\omega)$ はスペクトルとも呼ぶ．いま，$x(t)$ と $p_T(t)$ の積をとった信号 $x_S(t)$ は

$$x_S(t) = x(t) \cdot p_T(t) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT)$$

となる（下図参照）．続けて以下の問いに答えよ．



(ii) $e^{j\omega' t} \leftrightarrow 2\pi\delta(\omega - \omega')$ を利用して $p_T(t)$ のフーリエ変換 $P_T(\omega)$ を求めよ．

(iii) $x_S(t)$ のフーリエ変換 $X_S(\omega)$ を求めよ．

(iv) $x(t)$ が帯域制限されているとき，すなわち

$$X(\omega) = 0, \quad |\omega| > \omega_M$$

のとき，問い (iii) の $X_S(\omega)$ より $X(\omega)$ が正しく抽出される条件を求めよ．

論理回路・計算機システム

6．3つの2ビット入力 $a=(a_1,a_0)$，$b=(b_1,b_0)$，$c=(c_1,c_0)$ に対して，多数決の結果を $y=(y_1,y_0)$ として2ビット出力する「多数決結果出力回路」を考える．この多数決結果出力回路の各2ビット入力 a，b，c において，$(0,0)$ は入力組み合わせ禁止（ドントケア）となっており，各入力は (0,1)，(1,0)，あるいは (1,1) の内，いずれかの値を取る．一方，この多数決結果出力回路の2ビット出力 y は2ビット入力 a，b，c の内，2つ以上の入力が同じ値を取るときにはその値が出力され，それ以外の場合は $y=(0,0)$ が出力される．例えば $a=(1,1)$，$b=(1,0)$，$c=(1,0)$ のとき $y=(1,0)$ が出力され，$a=(1,1)$，$b=(1,0)$，$c=(0,1)$ のときは $y=(0,0)$ が出力される．この多数決結果出力回路を2ビットデコーダ D，多数決回路 M，2ビットエンコーダ E を接続することにより実現する．以下の問いに答えよ．

(i) 2ビットデコーダ D は2ビット入力 $p=(p_1,p_0)$ と3ビット出力 $q=(q_3,q_2,q_1)$ をもつ．ただし $p=(0,0)$ は入力組み合わせ禁止である．p の10進数表現が i $(i=1,2,3)$ のとき $q_i=1$ となり，それ以外の場合は0となる．すなわち D の入出力関係は次の表で与えられる．

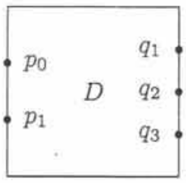
p_1	p_0	q_3	q_2	q_1
0	0	—	—	—
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

入力組み合わせ禁止に注意して，q_3，q_2，q_1 をなるべく簡単化された p_1，p_0 の論理式で表せ．

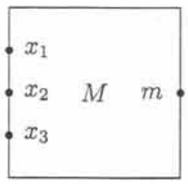
(ii) 多数決回路 M は3つの1ビット入力 x_1，x_2，x_3 と1ビット出力 m をもつ．m の値は x_1，x_2，x_3 の内，2つ以上の入力が1である場合は1，それ以外は0である．m を x_1，x_2，x_3 の最も簡単化された積和形で表せ．また，この多数決回路 M を NAND ゲートのみを用いて実現し，図示せよ．ただし，NAND ゲートの入力数に制限はないものとする．

(iii) 2ビットエンコーダ E は3ビット入力 $q=(q_3,q_2,q_1)$ と2ビット出力 $p=(p_1,p_0)$ をもつ．ただし，3ビット入力 q において複数の q_i が1となることはなく，これらは全て入力組み合わせ禁止とする．2ビット出力 p は $q_i=1$ のとき i の2進数表現となる．すなわち，$q_3=1$ ならば $p=(1,1)$，$q_2=1$ ならば $p=(1,0)$，$q_1=1$ ならば $p=(0,1)$ である．また，全ての $i=1,2,3$ に対して $q_i=0$ ならば $p=(0,0)$ となる．入力組み合わせ禁止に注意して，p_0，p_1 をなるべく簡単化された q_1，q_2，q_3 の論理式で表せ．

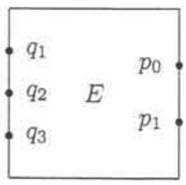
(iv) 3つの2ビットデコーダ D，3つの多数決回路 M，ならびに1つの2ビットエンコーダ E を接続して，この問題の最初で述べた，3つの2ビット入力 $a=(a_1,a_0)$，$b=(b_1,b_0)$，$c=(c_1,c_0)$ と2ビット出力 $y=(y_1,y_0)$ をもつ多数決結果出力回路を構成し，図示せよ．ただし，問い (i)，(ii)，(iii) で設計した3種の回路 D，M，E に対して，それぞれ下記の記号を用いること．



2ビットデコーダ D



多数決回路 M



2ビットエンコーダ E

基本アルゴリズム・プログラミング

7－1　2分探索木とは，2分木の各節点にデータを対応づけた上で，「ある節点のデータを x とするとき，その左部分木（左の子を根とする部分木）内のデータはすべて x より小さく，右部分木内のデータはすべて x より大きい」という条件を満たす2分木を指すものとする．この2分探索木について，以下の問いに答えよ．

(i) 高さ h の2分探索木において，節点数（ノード数）の最小値および最大値を答えよ．

(ii) 次の昇順に並んだ9個のデータについて，データ探索の計算量が最も少なくなる2分探索木（最適2分探索木）を一つ図示せよ．また，逆に計算量が最も多くなる2分探索木（最悪2分探索木）を一つ図示せよ．

(1,3,5,9,12,15,17,21,24)

(iii) プログラムAは，データ集合を順に2分探索木に収納する関数 insert と，あるデータ（ここでは「9」を設定）が2分探索木に含まれているかどうかを探索する関数 member を含むC言語のプログラムである．プログラムAにある空欄(1)～(3)を埋めよ．

(iv) プログラムAにおいて，問い(ii)で答えた最適2分探索木を構築するためには，プログラムAの＜α＞部分において配列 x[ ]をどのように初期化すべきか．問い(ii)で挙げた9個のデータの並び順を答えよ．

(v) 問い(iv)で初期化された配列 x[ ]に対してプログラム A を実行したとき，member 関数における printf 文の出力結果を全て示せ．ただし，プログラムAの実行が完了するまでの出力を順に示すこと．

<プログラム A>

```
#include <stdio.h>
#include <stdlib.h>
struct node{
  int element;
  struct node *left;
  struct node *right;
};

char member(int x, struct node *p)
{
   struct node *q;
   q=p;
   while(q != NULL)
   {
       if(q->element == x) return('Y'); /** Yes **/
       if(q->element < x)
            {printf("%2d : right¥n",q->element); q = q->right;}
       else
            {printf("%2d : left¥n",q->element); q = q->left;}
    }
   return('N'); /** No **/
}

void insert(int x, struct node **p)
{
   struct node *q, **r;
   r=p; q=*r;
   while(q != NULL)
   {
     if(q->element == x) return;
     if(q->element < x) { r=&(   (1)   ); q=q->right;}
     else { r=&(   (2)   ); q=q->left;}
   }
   *r = (struct node *)malloc(sizeof(struct node));
   (*r)->element =    (3)    ;
   (*r)->left = NULL;
   (*r)->right = NULL;
   if(*p == NULL) *p = *r;
   return;
 }

int main()
{
  struct node *init;
  int x[]={            <α>            };
  int i;   char a;
  init = NULL;
  for(i=0;i<9;i++) insert(x[i], &init);
  a=member(9, init);
  printf("(Y/N) = %c¥n",a);
  return(0);
}
```

7－2　プログラム B は，ある与えられた配列を前半部と後半部に分け，それぞれをソートした後，マージ（併合）することにより元の配列のソートを行うマージソートの C 言語プログラムである．以下の問いに答えよ．

（i）プログラム B にある空欄(1)～(3)を埋めよ．

（ii）プログラム B はデータを昇順にソートするプログラムであるが，一行を書き換えて降順にソートするプログラムに変更せよ．ただし，該当する一行の変更前と変更後を記せ．

（iii）プログラム B（昇順ソート）において，merge_main 関数における printf 文の出力結果をプログラム B の実行が完了するまで順に示せ．

＜プログラム B＞

```
#include  <stdio.h>
#include  <stdlib.h>
int    *buff;
void merge_main(int a[], int left, int right)
{
       int     i,j=0,k=left,p=0,center = (left + right) / 2;
       if (left < right) {
              merge_main(a,[    (1)    ], center);
              merge_main(a,[    (2)    ], right);
              for (i = left; i <= center; i++)
                   buff[p++] = a[i];/* a[]の前半部を buff[]へ代入 */

              while (i <= [    (3)    ] && j < p )
                   a[k++] = (buff[j] <= a[i]) ? buff[j++] : a[i++];
                                     /* a[]の後半部と buff[]を比較 */

              while ( j < p )
                   a[k++] = buff[j++]; /* buff[]の残りを a[]へ代入 */

              printf("L = %d  R = %d¥n",left,right);
       }
       return;
}
int mergesort(int a[], int n)
{
       if((buff = (int *)calloc(n,sizeof(int)))==NULL) return(-1);
       merge_main(a, 0, n - 1);
       free(buff);
       return(0);
}
int main()
{
       int     i,x[7]={34,23,54,12,3,56,78};
       int     nx = sizeof(x) / sizeof(x[0]);
       if(mergesort(x, nx) == -1){printf("Error"); return(1);}
       for (i = 0; i < nx; i++) printf("x[%d] = %d¥n", i, x[i]);
       return(0);
}
```