

受 験 番 号					

氏 名	

2024年度
放送大学大学院修士課程
文化科学研究科 文化科学専攻
情報学プログラム
筆記試験問題

試験日：2023年9月30日（土）

試験時間：9時30分～11時30分

注意事項

1. 試験開始の合図があるまで、この試験問題冊子は開かないでください。
2. 解答には、黒鉛筆かシャープペンシルを使用してください。
3. 配付されるものは、「試験問題冊子1冊」、「解答用紙5枚」及び「下書き用紙5枚」です。追加配付はしません。
4. 試験開始の合図の後、試験問題冊子を確認してください。試験問題冊子は、表紙、白紙、問題（11ページ）の順に綴じられています。試験問題冊子、解答用紙及び下書き用紙に落丁・過不足のある場合、あるいは印刷が不鮮明な場合には、手を挙げて試験監督員の指示に従ってください。
5. 試験問題冊子の所定欄に、受験番号及び氏名を記入してください。
6. 解答用紙は、「大問題（試験問題冊子に第1問、第2問…と表示されています。）」ごとに使用し、解答用紙の所定欄に、プログラム名、氏名、受験番号並びに「大問題」番号及び「大問題」ごとに何枚目であるかを、解答用紙別に必ず記入してください。
小問題及び選択問題を解答する際の番号等は、解答用紙のマス目の左側の「小問題番号等記入スペース」に記入してください。
なお、問題文中に別途記入方法の指示がある場合はそちらに従ってください。
7. 解答用紙1枚につき、800字まで記入することができます。解答用紙5枚のうち、情報学プログラムは4枚以内で解答してください。指定された字数に従って解答してください。
8. 試験問題冊子、解答用紙及び下書き用紙を綴じているホチキス針をはずしたり、中身を破り取ったりしてはいけません。
9. 試験問題冊子、解答用紙及び下書き用紙は試験終了後に回収します。試験問題冊子及び下書き用紙に解答を記入しても採点の対象にはなりませんので、必ず解答用紙に解答を記入してください。
10. 試験時間は2時間です。試験開始後40分を経過した後は、試験問題冊子、解答用紙及び下書き用紙を試験監督員に提出した上で、退室してもかまいません。ただし、試験終了5分前以降は退室できません。

情報学プログラム 筆記試験問題

問題は、第1問から第6問までである。

1. 第1問は共通問題である。全員が解答しなさい。
2. 第2問から第6問までは分野別問題である。出願時に提出した研究計画に最も近いと考えられる分野を1つだけ選び、その分野の問題に解答しなさい。なお、第2問はソフトウェア分野、第3問は情報基盤分野、第4問はヒューマン分野、第5問はマルチメディア分野、第6問は情報数理分野の問題である。

第1問 (全分野共通：必ず解答)

現代社会が抱える課題を1つ取り上げ、その課題を解決するために、どのように情報通信技術 (ICT) を活用できるかを論じなさい。ただし、下記のキーワードを2つ以上用いて800字以内で記述すること。なお、用いたキーワードには、下線を引くこと。

- DevOps
- SDGs
- インストラクショナルデザイン
- 画像生成 AI
- 技術者教育
- 教育データ
- 教育の情報化
- シンギュラリティ
- ソフトウェアの品質
- 大規模言語モデル
- デジタルツイン
- データ連携
- ブロックチェーン
- 分散型クラウド
- 倫理

第2問 ソフトウェア分野

以下の(1), (2)に答えよ。

(1) 次の説明を読み, (ア) ~ (ウ)に解答せよ。

T氏は, あるゲームのキャラクターの動作を制御するコントローラーの操作方法をM氏に伝えたい。このゲームでは, プレイヤーから見ると, 背景が左に流れていくので, 画面に向かって右向きにキャラクターが移動するように見える。コントローラーには「A」「B」「X」の3つのボタンがある。

表2.1に, A, B, X, Nの4つの指示と, それぞれの操作および意味を示す。

表2.1 指示・操作・意味

指示	操作	意味
A	「A」を1秒間押す。	キャラクターはジャンプする。
B	「B」を1秒間押す。	キャラクターは下にくぐる。
X	「X」を1秒間押す。	キャラクターの速度が上がる。
N	前の操作のあと, 1秒間何も押さないで待機する。	キャラクターはそのまま右に移動する。

この指示をつなげて, 指令(指示の列)を作る。例えば, 指令「XNANNNB」は, 以下の順に操作を行うという意味である。

- ボタン「X」を1秒間押して離し,
- 1秒間何も押さずに待機し,
- ボタン「A」を1秒間押して離し,
- 3秒間何も押さずに待機し,
- ボタン「B」を1秒間押して離す。

T氏からM氏に指令を伝えるには, 表2.2に示すルール1を用いて, 「0」「1」を利用したビット列に変換する。

表2.2 符号化のルール1

指示	A	B	X	N
符号	00	10	01	11

ここでは, ビット列で表した指令を「指令文」と呼ぶ。

「BAB」は100010という指令文になり、「NBXN」は①という指令文になる。一方、「②」は0110111111という指令文になる。表2.2の符号化のルール1は、どの指示も同じ長さの符号に変換する。指令文に含まれる指示1つあたりの符号の長さの平均（期待値）は、③である。

(ア) 空欄①～③を埋めよ。

実際の指令において、各指示の出現率は表2.3のとおりであった。

表 2.3 指示の出現率

指示	A	B	X	N
出現率	30%	20%	10%	40%

出現が多いNには短い符号を割り当てると、指令文が短くなることが期待できる。そこで、表2.4に示す新しい符号化のルール2を作成した。

表 2.4 符号化のルール2

指示	A	B	X	N
符号	10	100	000	1

ルール2を採用したとき、指令文に含まれる指示1つあたりの符号の長さの平均（期待値）は④であり、③より小さくなることから、指令文を短くできると考えた。

(イ) 空欄④を埋めよ。

ルール2をM氏に伝えたところ、M氏から「操作ができない」というクレームが届いた。そこで、表2.5に示す符号化のルール3を作成し、M氏に伝えた。

表 2.5 符号化のルール3

指示	A	B	X	N
符号	01	001	000	1

ルール3は、各指示に割り当てる符号の長さが同じであることから、指令文に含まれる指示1つあたりの符号の長さの平均（期待値）はルール2と同じであるが、M氏からは、「ルール2の問題点が改善された」という連絡があった。

(ウ) ルール 3 では、ルール 2 にあったどんな問題点が改善されたのか。情報学の用語などを利用して、300 字以内で説明せよ。

(2) 次の説明を読み、(ア) ~ (カ) に解答せよ。

本問では、3, 7, 1, 3 のような整数値の並びをリストと呼び、リストを構成する整数値をデータと呼ぶ。データの最大数を 6 とし、リストを管理するため、7 行 2 列の二次元配列 `slist` を用いる。`slist` の最初の行の行番号を 0、最初の列の列番号を 0 とし、行番号 i 、列番号 j の要素は `slist[i][j]` と表す。

二次元配列 `slist` は、次のように用いる。`slist` の行番号 1 以降の行の、列番号 0 の要素に、データを格納する。リストの先頭のデータが格納された行の行番号を `slist[0][1]` に格納する。データが格納された行の、列番号 1 の要素には、次のデータがあればそれが格納された行番号を、なければ 0 を格納する。リストにデータがない場合は、`slist[0][1]` に 0 を格納する。

例えば、`slist` の内容が図 2.1 の場合、`slist[0][1]` が 3 なので、リストの先頭のデータは `slist[3][0]` に格納された 3 である。`slist[3][1]` が 4 なので、次のデータは `slist[4][0]` に格納された 7 である。同様に、次のデータは `slist[1][0]` に格納された 1、その次のデータは `slist[5][0]` に格納された 3 である。`slist[5][1]` が 0 なので、次のデータはない。つまり、`slist` は、3, 7, 1, 3 というリストを格納している。なお、`slist[0][0]`、および `slist` の 2 行目と 6 行目の要素にも値が入っているが、これらはリストの管理には使われていない。

		列番号	
		0	1
行番号	0	0	3
	1	1	5
	2	5	1
	3	3	4
	4	7	1
	5	3	0
	6	4	2

図 2.1 `slist` の内容の例 1 (矢印は、説明のために追加した。)

(ア) `slist` の内容が図 2.2 の場合、`slist` が表すリストのデータを先頭から順に答えよ。

		列番号	
		0	1
行番号	0	0	5
	1	3	2
	2	5	4
	3	6	3
	4	6	0
	5	6	6
	6	2	1

図 2.2 slist の内容の例 2

slist は大域変数であり，関数から参照できる。また，擬似コードにおいて変数は整数を格納でき，宣言せずに使えるものとする。

リストのデータを，先頭から順に出力する関数 print_slist の実装を，擬似コード 1 に示す。

擬似コード 1

```
function print_slist()
  戻り値: なし
  -----
  pos ← slist[0][1]
  while (pos ≠ 0) {
    slist[pos][0] の値と改行を出力
    pos ← [ a ]
  }
```

(イ) 擬似コード 1 の [a] に入れる適切な式を答えよ。

リストに新たなデータを挿入する関数 insert の実装を，擬似コード 2 に示す。insert は，データ num を，行番号 base に格納されたデータの次に，つまり slist[base][1] 行目に挿入する。base が 0 の場合，num をリストの先頭に挿入する。insert が使う関数 free_pos は，slist の行番号 1 以降のすべての行がデータの格納に使われていれば 0 を返し，使われていない行があればそのうちのいずれかの行の行番号を返す。

擬似コード 2

```
function insert(num, base)
```

引数 num: 挿入するデータ

引数 base: 行番号 base に格納されたデータの次に num を挿入する
(base が 0 なら num をリストの先頭に挿入する)

戻り値: 成功したら true, 失敗したら false

```
pos ← free_pos()
if (pos = 0) {
    return false
}
slist[pos][0] ← num
slist[pos][1] ← [ b ]
slist[base][1] ← pos
return true
```

(ウ) 擬似コード 2 の [b] に入れる適切な式を答えよ。

関数 xyz の実装を, 擬似コード 3 に示す。

擬似コード 3

```
function xyz(num)
```

引数 num: 整数値

戻り値: なし

```
pos ← 0
while (slist[pos][1] ≠ 0) {
    if (slist[slist[pos][1]][0] = num) {
        slist[pos][1] ← slist[slist[pos][1]][1]
    } else {
        pos ← slist[pos][1]
    }
}
```

(エ) slist が図 2.2 の状態で, xyz(6) を呼び出した場合, 呼び出し後の slist[2][1] および slist[5][1] の値を答えよ。

(オ) 関数 xyz の役割を答えよ。

行番号 pos0 の行に格納されているデータの後ろには、2つ以上のデータが存在するものとする。行番号 pos0 の行に格納されているデータの次のデータと、その次のデータの順を入れ替えたい。その際、列番号 0 の要素の値は変更せず、列番号 1 の要素の値を変更して実現する。この処理の実装を、擬似コード 4 に示す。

— 擬似コード 4 —

```
// pos0 にはあらかじめ行番号が格納されている
pos1 ← slist[pos0][1]
pos2 ← slist[pos1][1]
slist[pos0][1] ← pos2
【 c 】
```

(カ) 擬似コード 4 の 【 c 】 に入れるべき 2 つの代入処理を擬似コードで示せ。

第3問 情報基盤分野

メタバース（インターネット上の仮想空間）の特徴と現在の事例について述べ、その長所と短所を取り上げた上で将来の有効な活用方法を述べよ。（1000字以内）

第4問 ヒューマン分野

ChatGPT を一例とする生成 AI について、その社会的普及がもたらす影響について、その利点と課題を、具体的な事例や研究結果を交えて記述してください。（1600字以内）

第5問 マルチメディア分野

マルチメディア技術は我々の生活に変化と多様性をもたらしている。あなたが着目するマルチメディア技術の事例を一つ取り上げ、その成果と、今後の課題について述べなさい。（800字以内）

第6問 情報数理分野

次の(1), (2)に答えよ。なお, 解答は結果だけを述べるのではなく, 結果に至る過程も述べる
こと。

(1) 以下の説明を読み, (ア), (イ), (ウ)に答えよ。以降, x_1, x_2, x_3, x, y はすべて, 0か1の
いずれかの値をとることとする。図6.1(i)は論理積を計算する素子である。すなわち, 図6.1(ii)
の真理値表が示すように, 入力 x_1, x_2, x_3 の値がすべて1のとき出力 y の値は1, それ以外のとき
出力 y の値は0となる。

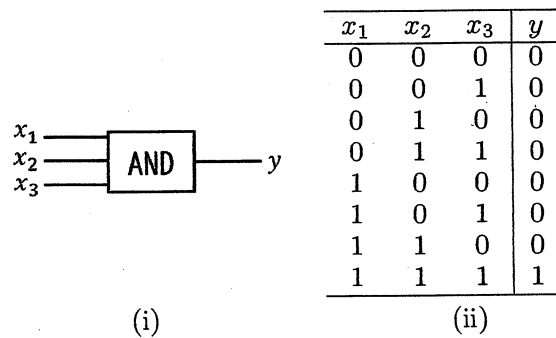


図6.1 (i) 3入力の論理積を計算する素子, (ii) 演算の真理値表

図6.2(i)は論理和を計算する素子である。すなわち, 図6.2(ii)の真理値表が示すように, 入力
 x_1, x_2, x_3 のうち1つでも値が1のとき出力 y の値は1, それ以外のとき出力 y の値は0となる。

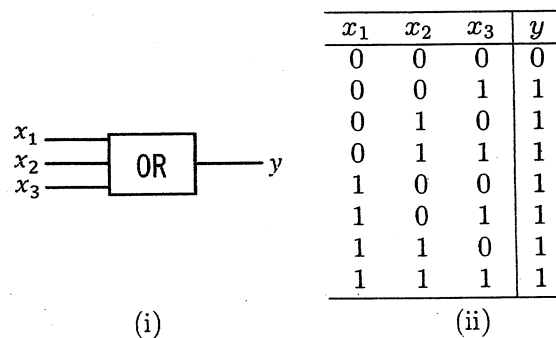


図6.2 (i) 3入力の論理和を計算する素子, (ii) 演算の真理値表

図6.3(i)は論理否定を計算する素子である。すなわち, 図6.3(ii)の真理値表が示すように, 入
力 x の値が0のとき出力 y の値は1, 入力 x の値が1のとき出力 y の値は0となる。

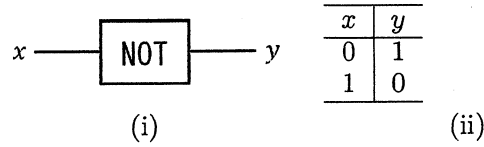


図 6.3 (i) 論理否定を計算する素子, (ii) 演算の真理値表

これらの論理演算素子を組み合わせて、様々な論理演算を行うことができる。例えば、図 6.4(i)で示される回路では、図 6.4(ii)の真理値表で示されるように、 x の値が0, 1のいずれでも出力 y の値は0となる。

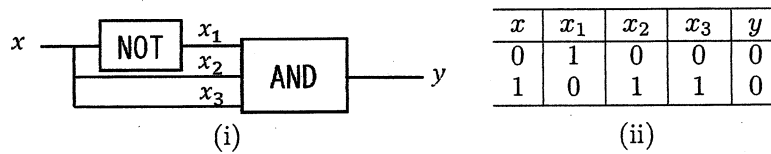


図 6.4 (i) 回路の例, (ii) 回路により実現される演算の真理値表

(ア) 図 6.2(i)の素子と図 6.3(i)の素子をそれぞれ必要な数だけ組み合わせて、図 6.1(ii)の真理値表で表される演算を行う回路を構成せよ。

(イ) 図 6.5(i)は否定論理和を計算する素子である。すなわち、図 6.5(ii)の真理値表が示すように、入力 x_1, x_2, x_3 の値がすべて0のとき出力 y の値は1, それ以外のとき出力 y の値は0となる。図 6.5(i)の素子を必要な数だけ組み合わせて、図 6.1(ii)の真理値表で表される演算を行う回路を構成せよ。

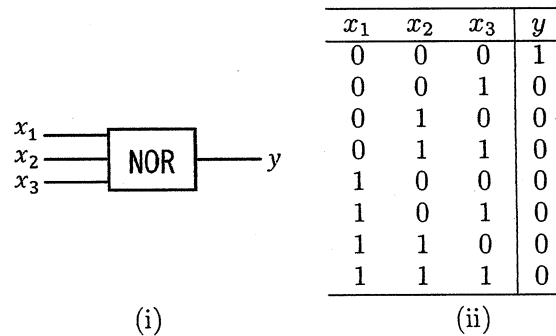


図 6.5 (i) 否定論理和を計算する素子, (ii) 演算の真理値表

(ウ) 論理積素子, 論理和素子, 論理否定素子を必要な数だけ組み合わせて、3つの入力 x_1, x_2, x_3 のうち2つ以上が1の値をとるときに出力 y の値が1, それ以外のときに出力 y の値が0となる回路を構成せよ。

(2) 以下の説明を読み, (ア) ~ (キ) に答えよ。1 回の試行で成功の事象 U が起こる確率を p とする。この試行を n 回繰り返して, U が起こる回数を X とする。 U が k 回起きたとき ($X = k$), その確率分布 Q は次式で表される。

$$Q(X = k) = {}_n C_k p^k (1 - p)^{n-k} \quad (6.1)$$

ただし, $k = 0, 1, 2, \dots, n$ であり, ${}_n C_k$ は n 個の異なるものの中から任意に k 個をとった組合せの数である。

(ア) $k {}_n C_k = n {}_{n-1} C_{k-1}$ を示せ。

(イ) $\sum_{k=0}^n Q(X = k)$ を求めよ。ただし, $(a + b)^n = \sum_{k=0}^n {}_n C_k a^k b^{n-k}$ である。

(ウ) X の期待値を求めよ。

(エ) Q を p の関数とする。 Q を対数変換し, 一次微分せよ。(ただし, $p \neq 0, 1$)

(オ) Q が極大となる確率 p を求めよ。

(カ) Q に対して $np = \lambda$ (一定) とし, $n \rightarrow \infty, p \rightarrow 0$ とする。このときの確率分布 R を求めよ。ただし, $\lim_{n \rightarrow \infty} \left(1 - \frac{\lambda}{n}\right)^n = e^{-\lambda}$ である。

(キ) 確率分布 R を λ の関数と考えて, R が極大となる λ を求めよ。ただし, $\lambda > 0, k > 0$ とする。