

令和 7 年度

専攻科入学者選抜
学力検査問題

専門 (情報コース)

(配点)

	出題分野	配 点
1	プログラミング・ ソフトウェア	80 点
2	計算機工学	60 点
3	情報数学	60 点

[注 意]

1. 問題は，指示があるまで開かないこと。
2. 問題用紙は，1 ページから 8 ページまでである。
検査開始の合図のあとで確かめること。
3. 答えは，すべて解答用紙に記入すること。

一関工業高等専門学校

1 (プログラミング・ソフトウェア)

問1 以下の設問それぞれに対し、選択肢から適切な答えを選び、記号で答えなさい。

- (1) オブジェクト指向における「オーバーライド」の説明として適切なものを答えなさい。
- (ア) 継承した親クラスで定義されたメソッドを子クラスで独自に定義して機能の上書きを行うこと。
 - (イ) クラス内に同じ名前だが引数の数や型が異なるメソッドを複数定義すること。
 - (ウ) 複数の親クラスを継承した子クラスを定義すること。
- (2) 「並列処理 (Parallelism)」の説明として適切なものを答えなさい。
- (ア) 複数の処理を交互に進める。シングルスレッドで処理する。
 - (イ) 複数の処理を交互に進める。マルチプロセスで処理する。
 - (ウ) 複数の処理を同時に進める。マルチコア・マルチプロセスで処理する。
 - (エ) 複数の処理を同時に進める。マルチコア・マルチスレッドで処理する。
- (3) マルチスレッド処理の特徴として適切なものを答えなさい。
- (ア) スレッドごとにアドレス空間が必要である。
 - (イ) スレッドは単一のアドレス空間を共有している。
- (4) コンパイラ方式のプログラミング言語における「リンカー」の役割として適切なものを答えなさい。
- (ア) ソースコードを機械語に変換する。
 - (イ) オブジェクトコードをライブラリと紐付けてロードモジュールを作成する。
 - (ウ) ロードモジュールを主記憶の空き容量に読み込む。

問2 (図2-1) はリストを二つの1次元配列で実現したものです。配列要素 $item[i]$ と $next[i]$ の対がリストの一つの要素に対応し、 $item[i]$ に要素の値が入り、 $next[i]$ に次の要素の番号が入ります。また、 $next[0]$ がリストの先頭(1番目)の要素を指し、 $next[i]$ の値が0である要素はリストの最後を示すものとします。ただし、ここではリストに連結されていない要素に対応する $next[i]$ は初期化を行い忘れたためデタラメな値となっています。この時、以下の設問に答えなさい。

	0	1	2	3	4	5	6	7	8	9
item		A	B	C	D	E	F	G	H	I
	0	1	2	3	4	5	6	7	8	9
next	2	1	7	0	9	4	3	3	2	5

図2-1. リストを構成する2配列

- (1) 要素数を n とした時、このリストの要素へのランダムアクセスにかかる時間計算量をオーダー記法で答えなさい。
- (2) リストの2番目と3番目との間に値がDである要素を挿入したときのリストの要素を順番通りに答えなさい。
- (3) (2) 下線の処理を行った後の、配列 $next$ の正しい状態を答えなさい。ただし、この際にリストに連結されていない要素に対応する $next[i]$ は値が削除されたものとします。

問3 Python 実装（プログラム3－1）は幅優先探索を実装したプログラムです。ここでは数字のみを要素として持つリストから木を作成するために TreeNode クラスと make_tree 関数を、また幅優先探索の実装として search 関数を用意しました。これを元に以下の問いに答えなさい。

1	class TreeNode:
2	def __init__(self, val:int, left=None, right=None):
3	self.val = val
4	self.left = left
5	self.right = right
6	def __repr__(self) -> str:
7	return f"{self.val}"
8	
9	def make_tree(tree:list[int], node, i:int, n:int):
10	# i := current target node number. n := number of nodes.
11	if ①:
12	if tree[i] is None: return None
13	node = TreeNode(tree[i])
14	node.left = make_tree(tree, node.left, 2*i+1, n)
15	node.right = ②
16	return node
17	
18	def search(root:TreeNode, target:int=5):
19	path = []
20	X = [root]
21	while X:
22	print(X) # logging
23	node = X.pop(③)
24	path.append(node)
25	if target == node.val: break
26	if node.left: X.append(node.left)
27	if node.right: X.append(node.right)
28	else: print(f"{target} does not exist in this tree .")
29	return path
30	tree = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
31	root:TreeNode = make_tree([1, 2, 3, 4, 5, 6, 7, 8, 9, 10], None, 0, len(tree))
32	print("探索中のXの様子:")
33	path = search(root, 6)
34	print("経路:")
35	print(path)

プログラム3－1. bfs.py

(1) 以下の選択肢から に適切なものを選び、記号で答えなさい。

- (ア) $n > i$
- (イ) $n \geq i$
- (ウ) $n \leq i$
- (エ) $n < i$
- (オ) $n == i$

(2) 以下の選択肢から に適切なものを選び、記号で答えなさい。

- (ア) `make_tree(tree, node.right, 2*i+1, n-1)`
- (イ) `make_tree(tree, node.right, 2*i+2, n-1)`
- (ウ) `make_tree(tree, node.right, 2*i+1, n)`
- (エ) `make_tree(tree, node.right, 2*i+2, n)`

(3) 以下の選択肢から に適切なものを選び、記号で答えなさい。

- (ア) 0
- (イ) 1
- (ウ) -1
- (エ) n

(4) `search` 関数を修正して行きがけ順の深さ優先探索を実装します。このときに以下の選択肢から に適切なものを選び、記号で答えなさい。

- (ア) 0
- (イ) 1
- (ウ) -1
- (エ) n

(5) 枝の数を $|E|$ 、ノードの数を $|V|$ とした場合の、幅優先探索の時間計算量をオーダー記法で答えなさい。

(6) 枝の数を $|E|$ 、ノードの数を $|V|$ とした場合の、幅優先探索の空間計算量をオーダー記法で答えなさい。

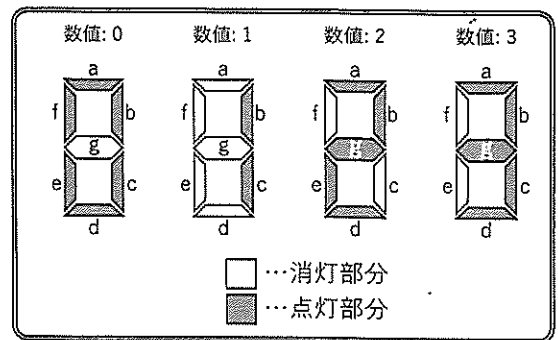
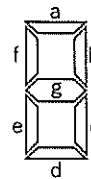
(7) (1)(2)(3)において適切に修正された(プログラム3-1)で作成される木を図示しなさい。

(8) (1)(2)(3)において適切に修正された(プログラム3-1)の35行目に対応する標準出力を示しなさい。

(9) (1)(2)(4)において適切に修正された(プログラム3-1)の35行目に対応する標準出力を示しなさい。

2 (計算機工学)

問1 (図2-1)に示すような7セグメントLEDに、組み合わせ論理回路を用いて0~3の数値を表示する。数値は2進数で入力X,Yに与え、a~gへの出力が"1"の時にそれぞれのセグメントが点灯するものとする。("0"の時は消灯)



(図2-1) 7セグメントLEDの点灯パターン

(1) (表2-1)に示す真理値表を完成しなさい。

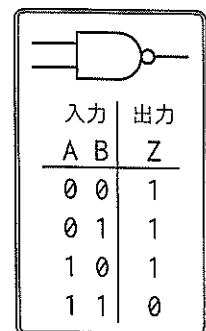
(表2-1) 真理値表

数値	入力		出力						
	X	Y	a	b	c	d	e	f	g
「0」	0	0							
「1」	0	1	0	1	1	0	0	0	0
「2」	1	0							
「3」	1	1							

(2) a, b, c, e, f, g の論理式を求め、選択肢(ア)~(タ)から選んで記号で答えなさい。(※同じ選択肢を複数回答しても良い。) なお、 $\bar{X}$ は X の否定(NOT), 「 $\cdot$ 」は論理積(AND), 「+」は論理和(OR)を表すものとする。

(ア) X	(オ) $X \cdot Y$	(ケ) $X + Y$	(ス) $X \cdot Y + \bar{X} \cdot \bar{Y}$
(イ) $\bar{X}$	(カ) $X \cdot \bar{Y}$	(コ) $X + \bar{Y}$	(セ) $\bar{X} \cdot Y + X \cdot \bar{Y}$
(ウ) Y	(キ) $\bar{X} \cdot Y$	(サ) $\bar{X} + Y$	(ソ) 0
(エ) $\bar{Y}$	(ク) $\bar{X} \cdot \bar{Y}$	(シ) $\bar{X} + \bar{Y}$	(タ) 1

(3) (表2-1)に示す出力 d の論理式は $d = X + \bar{Y}$ である。この論理式を実現する論理回路を、(図2-2)に示す2入力の NAND 素子のみを用いて構成し、図示しなさい。



(図2-2) 2入力のNAND素子

問2 次の論理式について答えなさい。

$$Z = \bar{A} \cdot \bar{B} \cdot C \cdot D + \bar{A} \cdot \bar{B} \cdot C \cdot \bar{D} + \bar{A} \cdot B \cdot C + A \cdot C + A \cdot \bar{C} \cdot D$$

(1) Z に関するカルノー図を作成しなさい。

(2) 作成したカルノー図を参考に、最も簡単化した加法形の論理式として Z を示しなさい。

問3 10進数で0～9までカウントすることの出来るアップカウンタをSR-FFで構成する場合、SR-FFは最低何個必要か答えなさい。

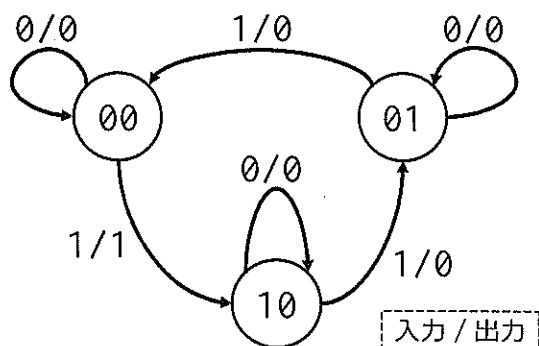
問4 (図2-3)に示す状態遷移図に基づき、JK-FFを用いた順序回路を設計することを考える。各状態は、2つのJK-FFの出力 Q_2, Q_1 を用いて(表2-2)のように割り当てるものとする。なお、使用するJK-FFはポジティブエッジ動作型とし、クロック信号 CLK を入力するものとする。

(1) この順序回路の状態遷移表(表2-3)の「(1)次の状態、出力」部分を完成しなさい。(※ドントケア項は $\times$ で示すこと。)

(2) 同様に、この順序回路の状態遷移表(表2-3)の「(2)所要入力」部分を完成しなさい。(※ドントケア項は $\times$ で示すこと。) 所要入力とは、現在の状態から次の状態に遷移させるために必要な入力であり、 $Q_2 \rightarrow Q_2^+$ に遷移させるために必要な入力は J_2, K_2 に、 $Q_1 \rightarrow Q_1^+$ に遷移させるために必要な入力は J_1, K_1 に記入するものとする。

(3) この順序回路に、タイミングチャート(図2-4)に示す入力 A およびクロック信号 CLK を各JK-FFに与えた場合、各FFの状態 Q_2, Q_1 および出力 Z はどのように変化するか。

初期状態を $Q_2 = Q_1 = 0$ として、(図2-4)のタイミングチャートを完成しなさい。



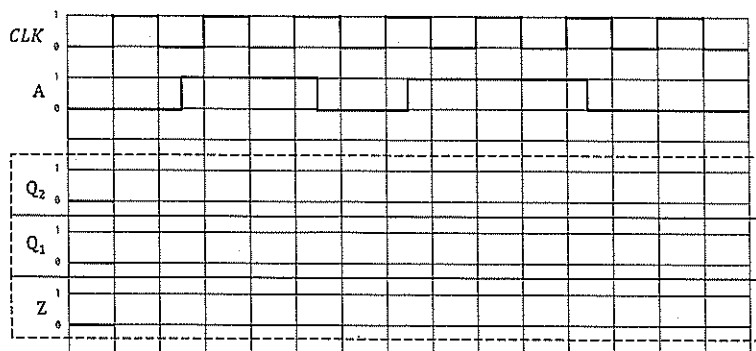
(図2-3) 状態遷移図

(表2-2) 状態の割当

状態	Q_2	Q_1
「00」	0	0
「01」	0	1
「10」	1	0

(表2-3) 状態遷移表

現在の状態 Q_2 Q_1	入力 A	(1) 次の状態 出力		(2) 所要入力			
		Q_2^+	Q_1^+	Z	J_2	K_2	J_1 K_1
0 0	0						
0 0	1						
0 1	0						
0 1	1						
1 0	0						
1 0	1						
1 1	0						
1 1	1						



(図2-4) タイミングチャート

3 (情報数学)

問1 以下の各問いに答えなさい。

- (1) 集合 $\{(x, y) \in \mathbb{Z}^2 \mid x^2 + 4y^2 \leq 4\}$ の外延的表記(要素を具体的に列挙する書き方)を答えなさい。
- (2) 土曜日の 5^{500} 日後は何曜日であるか, 答えなさい。

問2 鳩の巣原理とは, 「 n 個の物を m 個の箱に入れるとき, $n > m$ であれば, 少なくとも1個の箱には1個より多い物が入ることになる」という原理である。ディリクレの箱入れ原理, 部屋割り論法などとも呼ばれる。この原理を用いて以下の各問いに答えなさい。

- (1) 10名以上の児童が在籍するような6年制の小学校について考える。この小学校から児童を無作為に複数名選出したとき, その選出された児童の中の少なくとも2名が同じ学年に属するようにしたい。例えば3名選出してその内訳が「2年生, 5年生, 5年生」であれば5年に属する者が2名いるので条件を満たしている。しかし「1年生, 2年生, 6年生」のような選出の場合は条件を満たさない。各学年の児童数が不明な場合に, どんな選び方をしても「同学年が2名以上」という条件が満たされるようにするには, 最低限何名を選出すればよいか。最低限必要な選出人数の数値を答えなさい。
- (2) 各学年3クラス, 各クラスは30名で構成される6年制の小学校について考える。この小学校に在籍する児童の中に, 誕生日が同一の児童が2名以上いる確率は何%であるか, 答えなさい。

問3 次の文章の各空欄に当てはまる数あるいは式を答えなさい。同一の数や式を複数回使用することもあり得る。また, 同一番号の空欄が複数回現れることもあるため, 番号に注意すること。

1000 円未満の金額を硬貨によって支払う状況を考える。硬貨としては 1 円硬貨, 5 円硬貨, 10 円硬貨, 50 円硬貨, 100 円硬貨, 500 円硬貨の 6 種類があるものとする。ある金額を支払う場合に, 硬貨の組み合わせは複数あり得る。硬貨の総数が最小となるような組み合わせでは何枚の硬貨が必要となるかを考える。

[組み合わせベクトルの定義]

以下では, k 円硬貨の枚数を c_k として作られる6次元ベクトル $c = (c_1, c_5, c_{10}, c_{50}, c_{100}, c_{500})$ を用いて硬貨の組み合わせを表すこととし, このベクトルを組み合わせベクトルと呼ぶ

ことにする。組み合わせベクトルの要素 c_k は非負の整数である。また組み合わせベクトルの全体集合を U とする。例えば30円支払う場合には $(30, 0, 0, 0, 0, 0)$ (1円硬貨30枚) という支払い方や $(0, 2, 2, 0, 0, 0)$ (5円硬貨2枚と10円硬貨2枚) という支払い方が存在する。

[関数の定義]

ある組み合わせベクトル c についての合計金額を表す関数を $h(c)$ とし、硬貨の総数を表す関数を $f(c)$ とする。定義より $c = (0, 1, 1, 0, 0, 0)$ のとき $h(c) = 15$ かつ $f(c) = 2$ となる。また $c = (25, 1, 0, 0, 0, 0)$ とすると $h(c) = \boxed{(1)}$ かつ $f(c) = \boxed{(2)}$ となる。この関数 h を用いれば n 円の金額を表す組み合わせベクトル全体からなる集合 $S(n)$ は以下のように表せる。

$$S(n) = \{c \in U \mid \boxed{(3)}\}$$

さらに関数 f と集合 $S(n)$ を用いれば、ある金額 n を表すために必要な硬貨の最小枚数 $q(n)$ は以下のような数式で表せる。

$$q(n) = \min\{\boxed{(4)}\}$$

以下ではこの $q(n)$ の計算方法について考える。

[$S(n)$ の性質]

- 定義から、以下の式が成り立つ。

$$S(30) \supset \{(30, 0, 0, 0, 0, 0), (20, \boxed{(5)}, 0, 0, 0, 0), (5, \boxed{(6)}, 0, \boxed{(7)}, 0, 0)\}$$

- 集合 $S(n)$ は必ず1つ以上の組み合わせベクトルを含む(性質A)。 n を用いて表される組み合わせベクトル $c = \boxed{(8)}$ は $S(n)$ に含まれる。
- $S(10)$ は4つの組み合わせベクトルから成る。それらのベクトルを、硬貨の総数の多いものから順に並べると $\boxed{(9)}$, $(5, 1, 0, 0, 0, 0)$, $\boxed{(10)}$, $\boxed{(11)}$ である。

[$h(c)$ と $f(c)$ の式]

$c = (c_1, c_5, c_{10}, c_{50}, c_{100}, c_{500})$ とする。 $h(c)$ 及び $f(c)$ を c の各要素を用いて表した式は以下のようになる。

$$h(c) = \boxed{(12)}, \quad f(c) = \boxed{(13)}$$

[硬貨総数が最小となるような組み合わせベクトル]

$S(n)$ は有限集合であり、前述の性質Aの通り1つ以上の要素を含む。したがって $S(n)$ に含まれる組み合わせベクトル c の中で、 $f(c)$ が最小となるベクトルが必ず存在する。そのうちのひとつをここでは $g = (g_1, g_5, g_{10}, g_{50}, g_{100}, g_{500}) \in S(n)$ とする。このようにして定義

した g の各要素 g_k ($k \in \{1, 5, 10, 50, 100, 500\}$)の上限を求めることを考える。

まずは g_1 について考える。ある組み合わせベクトル $c = (c_1, c_5, c_{10}, c_{50}, c_{100}, c_{500}) \in S(n)$ を考えて、 $5 \leq c_1$ と仮定する。このとき新たな組み合わせベクトル r を

$$r = (c_1 + \boxed{(14)}, c_5 + 1, c_{10}, c_{50}, c_{100}, c_{500})$$

のように定義すれば $h(c) = h(r)$ が成り立ち、金額としては同額となる。このとき、組み合わせベクトル c, r の硬貨の総数について比較すると $f(r) = f(c) - \boxed{(15)}$ となる。

したがって、 $5 \leq c_1$ となるような組み合わせベクトル c は硬貨の総数が最小にはならない ($f(c) \neq q(n)$)。つまり逆に、硬貨の総数が最小となるような組み合わせベクトル g については $0 \leq g_1 < \boxed{(16)}$ が成り立つ。 g_{10}, g_{100} についても同様の議論が成り立つので、 $k \in \{1, 10, 100\}$ について $0 \leq g_k < \boxed{(16)}$ となることが分かる。

g_5, g_{50}, g_{500} についても同様の手順で考えれば $k \in \{5, 50, 500\}$ について $0 \leq g_k < \boxed{(17)}$ となることが分かる ($n < 1000$ であることに注意)。つまり以下の式が成り立つ。

$$0 \leq g_k < \boxed{(16)} \quad (k \in \{1, 10, 100\}), \quad 0 \leq g_k < \boxed{(17)} \quad (k \in \{5, 50, 500\}) \quad \textcircled{1}$$

これらの上限から、組み合わせベクトル g の各要素の値を求めることができる。

まず $g \in S(n)$ であるから $n = h(g)$ であり、これと $0 \leq g_1 < \boxed{(16)}$ を合わせて考えれば $g_1 = n \bmod \boxed{(18)}$ であることが分かる。このようにして求めた g_1 を用いれば

$$g_5 = \frac{n - g_1}{\boxed{(19)}} \bmod \boxed{(20)}$$

のようにして g_5 を求めることができる。 $g_{10}, g_{50}, g_{100}, g_{500}$ についても同じような手順で順々に求めることができる。このようにして求めた g を用いれば $q(n) = f(g)$ であるので、 $q(n)$ が計算できる。

[必要硬貨総数の平均値]

0円から999円までのそれぞれの金額 n について、その金額を表す硬貨の組み合わせで硬貨総数の最小値 $q(n)$ を考え、それらの平均値を u とする。 u は以下の式で表される。

$$u = \frac{1}{1000} \sum_{n=0}^{999} q(n)$$

具体的な数値を求めると $u = \boxed{(21)}$ となる (前述の式①を活用すると計算しやすい)。

硬貨の種類が1円硬貨、10円硬貨、100円硬貨の3種類であった場合について考える。その場合の0円から999円までのそれぞれの金額について、最低限必要な硬貨の枚数を考えてその平均値を求めると $\boxed{(22)}$ 枚となる。