令和８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９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　プログラミングに関する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）　カプセル化に関する説明として，最も適切なものを答えなさい。

（ア）データとそのデータに対する手続きをひとつにまとめ，オブジェクトの内部を隠ぺいすること。

（イ）ある上位のクラスをもとに下位クラスを定義し，下位クラスに上位クラスの属性を引き継がせること。

（ウ）ある実体としてのオブジェクト群から共通する性質・要素・動作を引き出して，それらの事実を包括する概念を定義すること。

（エ）一般化されたクラスに，ある実体が持つ固有の性質・要素・動作を加えることで，その実体に特化した概念を定義すること。

（２）　シャローコピー（shallow copy）とディープコピー（deep copy）に関する説明として，最も適切なものを答えなさい。

（ア）シャローコピーは新しいメモリ領域を割り当て，ディープコピーは元のオブジェクトと同じメモリ領域を共有する。

（イ）シャローコピーは元のオブジェクトと同じメモリ領域を共有し，ディープコピーは新しいメモリ領域を割り当てる。

（ウ）シャローコピーもディープコピーも新しいメモリ領域を割り当てる。

（エ）シャローコピーもディープコピーも元のオブジェクトと同じメモリ領域を共有する。

（３）　ポリモーフィズムを実現するために使われる ABC（抽象基底クラス）と Protocol（ダックタイピング）の思想の違いについて着目し，それぞれに対応している説明を一つずつ答えなさい。

（ア）A 型を継承しているならば B 型である。

（イ）A 型を継承していないならば B 型である。

（ウ）A ができるならば B 型である。

（エ）A ができないならば B 型ではない。

（４）　以下の Python コードでは以下の数式を実装しています。しかしこれをそのまま利用すると，オーバーフローが発生する可能性が高いという問題があります。この対策としてコード中の $\boxed{①}$ に対して最も適切なものを答えなさい。

$$y_k = \frac{\exp(x_k)}{\sum_{k'=1}^{K} \exp(x_{k'})}$$

```
import numpy as np
def softmax(x, axis=-1): # x はデータ数 n×特徴量数 d の配列。dtype は np.float32。
    x = x - x.[ ① ](axis, keepdims=True)
    exp_x = np.exp(x)
    return exp_x / exp_x.sum(axis, keepdims=True)
```

プログラム１－１：softmax.py

（ア）max（イ）min（ウ）mean（エ）std

（5） 変数のスコープを確認するために以下の Python コードを作成しました。このプログラムを実行した際に表示される文字列として正しいものを以下の選択肢中から一つ選びなさい。

```
x = 1
def A(x):
    x = x+10
    def B():
        global x
        x = x+100
        return x
    print("①", x, end=" ")
    return B
print("②", A(5)(), end=" ")
print("③", x, end=" ")
```

プログラム 1 − 2 : scope.py

(ア) ① 101 ② 111 ③ 1
(イ) ① 101 ② 101 ③ 101
(ウ) ① 15 ② 101 ③ 101
(エ) UnboundLocalError: cannot access local variable 'x' where it is not associated with a value

問2 k-近傍法によるクラス分類器を用いたクラスラベル推論関数の疑似コードを以下に示します。ただし，訓練データの数を n，特徴量の数を d とし，ソートアルゴリズムとしてクイックソートを用いることとします。これ参考にして，推論における時間計算量（距離計算とソートの両方を含めた時間計算量）を平均計算量でオーダー記法を用いて答えなさい。

```
Input:
  - Query point: q (a 1-dimensional array with d elements)
  - Training data points: X = {x_1, x_2, ..., x_n} (each x_i is a 1-dimensional array with d elements)
  - Number of neighbors: k

For each training data point x_i in X:
    Calculate the distance between q and x_i
End For
Sort the distances in ascending order
Select the k nearest neighbors
Determine the majority class label among the k nearest neighbors
Return the majority class label
```

プログラム 1 − 3 : k-近傍法の疑似コード

問3　以下のプログラムはダイクストラ法を実装しています。これをもとに以下の問に答えなさい。

```
class Node:
    def __init__(self, char: str, freq: int):
        self.char = char
        self.freq = freq

def dijkstra(graph, start):
    distances = {node: float('inf') for node in graph}
    distances[start] = 0
    path = {node: None for node in graph}
    nodes = [Node(node, distances[node]) for node in graph]
    print("Initial distance:", distances)
    print("Initial path:    ", path)
    i = 0
    while nodes:
        nodes.sort(key=lambda x: x.freq) # Timsort, 最悪計算量 O(n log n)
        current_node = [ ① ]
        current_char = current_node.char
        print(f"({i}) Current node: {current_char}, Distance: {distances[current_char]}")
        for neighbor, weight in graph[current_char].items():
            distance = distances[current_char] + weight
            if distance [ ② ] distances[neighbor]:
                distances[neighbor] = distance
                path[neighbor] = current_char
                for node in nodes:
                    if [ ③ ]:
                        node.freq = distance
                        break
        i += 1
        print(f"({i}) Temp distance:", distances)
        print(f"({i}) Temp path:    ", path)
    return distances, path

# Example usage:
graph = {
    'A': {'B': 1, 'C': 4},
    'B': {'A': 1, 'C': 2, 'D': 5},
    'C': {'A': 4, 'B': 2, 'D': 1},
    'D': {'B': 5, 'C': 1}
```

```
}
distances, path = dijkstra(graph, 'A')
print("Shortest distances from start node:", distances)
print("Shortest path tree:", path)
```

プログラム１－４：dijkstra.py

（１）［ ① ］に対応するコードを以下の選択肢から一つ答えなさい。

（ア）nodes[0] # nodes の最初の要素を取り出す。nodes からその要素は削除されない。

（イ）nodes[-1] # nodes の最後の要素を取り出す。nodes からその要素は削除されない。

（ウ）nodes.pop(0) # nodes の最初の要素を取り出す。nodes からその要素は削除される。

（エ）nodes.pop(-1) # nodes の最後の要素を取り出す。nodes からその要素は削除される。

（２）［ ② ］に対応する等号・不等号を答えなさい。

（３）［ ③ ］に対応するプログラムを答えなさい。

（４）プログラムが全て穴埋めされた状態であるとき，このプログラムを実行した際の標準出力を解答欄の下線部を補完する形で答えなさい。<u>ただし，解答欄記入の際にはシングルクォーテーションマークやダブルクォーテーションマーク，及び None は省略して良いものとする。</u>

以下の（５）では，非負の重み付きグラフG，グラフGの頂点集合V，グラフGの辺集合Eとして回答すること。

（５）ノードを保存するデータ構造としてヒープを用いるダイクストラ法の実装の最悪時間計算量をオーダー記法で答えなさい。但し，ヒープへの要素の追加とヒープからの最小値の取得・削除は $O(\log|V|)$で行えるものとする。

2 (計算機工学)

問１　次の問いに答えなさい。

（１）2進数 1011_2 を10進数で表しなさい。(例 10進数の13ならば 13_{10} で表される)

（２）10進数 13_{10} を2進数で表しなさい。(例 2進数の1011ならば 1011_2 で表される)

（３）2つの2進数 1010_2 と 1101_2 に対して，次の演算を行い，結果を2進数で答えなさい。

（ア）論理積（AND）

（イ）論理和（OR）

（ウ）排他的論理和（XOR）

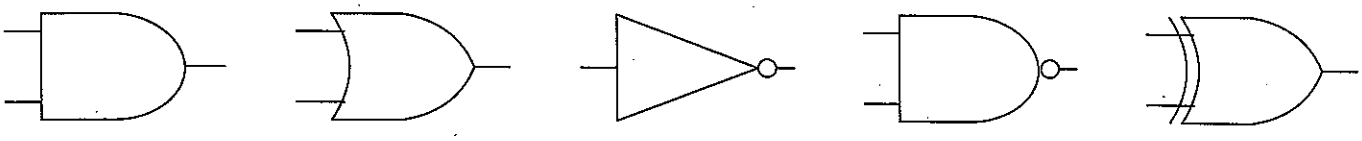
問２　（表２−１）の真理値表は，2つの入力a，bに対するある論理回路の出力outを示したものである。このとき，以下の問いに答えなさい。

（表２−１）真理値表

a	b	out
0	0	0
0	1	1
1	0	1
1	1	0

（１）この真理値表の出力outに対応する加法標準形（SOP: Sum of Products）を論理積（AND），論理和(OR)，論理否定（NOT）を用いた論理式で表しなさい。(ただし，論理記号は次のとおりとする。AND「·」, OR「+」，NOT「-」（aの論理否定の場合$\bar{a}$））

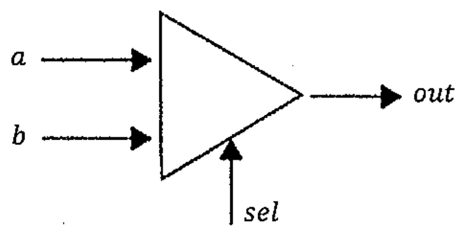
（２）（１）で導出した論理式をもとに, 2 入力 AND 回路, 2 入力 OR 回路, NOT 回路を用いて, 最小構成の論理回路として図示しなさい。(なお，（図２−１）に示す記号を用いて描くこと。)



（図２−１）左から AND,　OR,　NOT,　NAND,　XOR 回路

（３）（２）で作成した論理回路を, 2 入力 NAND 回路のみを用いて図示しなさい。(なお，（図２−１）に示す記号を用いて描くこと。)

問3　マルチプレクサは，複数の入力から1つの値を選択するスイッチ回路として，CPUや通信制御などで広く用いられる。(図2−2)は，2つのデータビット入力a, bと，1つの選択ビットselを持つ2入力1セレクタのマルチプレクサである。(図2−2)の真理値表を(表2−2)に示す。このマルチプレクサは，(表2−2)の選択ビットselに応じて，入力aまたはbのいずれかを出力outとして出力する。



(図2−2)　マルチプレクサ

(表2−2) マルチプレクサの真理値表

sel	out
0	a
1	b

このとき，以下の問いに答えなさい。

(1) このマルチプレクサの出力outを，a, b, sel を入力として，論理積(AND)，論理和(OR)，論理否定(NOT)を用いた論理式で表しなさい。(ただし，論理記号は次のとおりとする。AND「・」, OR「+」，NOT「-」(aの論理否定の場合$\bar{a}$))

(2)(1)の論理式を論理回路として図示しなさい。
(なお，(図2−1)に示す記号を用いて描くこと。)

問4　半加算器(Half Adder)は，2つの1ビット入力aとbを加算し，和(sum)と桁上がり($carry$)を出力する論理回路である。これは，全加算器(Full Adder)や加算回路の基本構成ブロックとして重要な役割を果たす。この回路の真理値表は(表2−3)のとおりである。

(表2−3) 半加算器の真理値表

a	b	sum	$carry$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

このとき，以下の問いに答えなさい。

(1) この真理値表の出力sumを，乗法標準形を用いた論理式で，$carry$を論理積(AND)で表しなさい。(ただし，論理記号は次のとおりとする。AND「・」, OR「+」，NOT「-」(aの論理否定の場合$\bar{a}$))

(2)(1)の論理式を，論理回路として図示しなさい。
(なお，(図2−1)に示す記号を用いて描くこと。)

(3)(2)を再構成し，2入力ANDを2個，2入力ORおよびNOTを1個ずつ用いた最小構成の論理回路として図示しなさい。(なお，(図2−1)に示す記号を用いて描くこと。)

$\boxed{3}$（情報数学）

問１ 2001年1月1日は月曜日であった。2004年，2008年などは閏年で年間の日数は366日であり，閏年でない年は年間の日数は365日である。これを踏まえて以下の各問いに答えなさい。

(1) 2002年1月1日は何曜日であるか，答えなさい。

(2) 1月1日が月曜日になる年のうち，2002年以降の年で2001年に最も近い年を答えなさい。

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) 令和6年5月1日時点での一関高専未来創造工学科の在籍学生数は774名である。誕生日が同じである学生が存在する確率は何%であるか，答えなさい。

(2) 1万人の人から無作為に何人かを選出することとする。選出された人の中に血液型(A，B，AB，Oの4種類)と誕生日(1月1日から12月31日)のどちらもが互いに同じとなるような2名の人が必ず含まれるようにするには，最低何人選出すれば良いか答えなさい。但し，選出候補の1万人の中には閏年の2月29日生まれの人はいないものとする。

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。

1対1で対戦するゲームについて大会を開く際，試合のために必要となる時間を見積もりたい。大会とゲーム，及び問題文や解答内で使用する関数については以下のようなものとする。

- ゲームは参加者2名が1対1で対戦する形式であり，1試合当たりの所要時間は1時間である。試合の結果は一方の参加者が勝ち，もう一方の参加者は負けとなる（引き分けはない）。
- 大会は，試合のたびにその敗者を除外していく勝ち残り式のシングルイリミネーション方式トーナメントである。最後に残る1名の参加者が決定した時点で大会は終了とし，その残った参加者を優勝者とする。
- 大会の参加者数をN名とする。
- 大会の総試合数は参加者数によって定まるのでその数を$f(N)$とする。

- 実数xについて，「x以下の整数のうち最大のもの」を$\lfloor x \rfloor$の記号で表す。
- 実数xについて，「x以上の整数のうち最小のもの」を$\lceil x \rceil$の記号で表す。

参加者数がN名の場合の大会の総試合数$f(N)$について考える。$N=2$の場合，1試合で優勝者が決定できるので$f(2)=1$となる。$N=3$の場合を考えると$f(3)=\boxed{(1)}$であり，同様に考えれば$f(8)=\boxed{(2)}$となる。1試合ごとにその敗者1名が除外されていくことを考えると，優勝者1名が定まるまでには$\boxed{(3)}$名が除外されているはずなので$f(N)=\boxed{(3)}$となる。

次に，参加者数がN名の大会を終了するのにかかる最短所要時間について考える。1試合行うたびにその敗者1名が除外されるので，大会全体の所要時間を最短とするにはなるべく多くの試合を同時に実施すればよい。例えば残っている参加者が5名という状態であれば最大で$\boxed{(4)}$個の試合を同時に行うことができ，それらの試合後の残り参加者数は$\boxed{(5)}$名となる。残っている参加者が8名の場合には最大で$\boxed{(6)}$個の試合を同時に行うことができそれらの試合後の残り参加者数は$\boxed{(7)}$名となる。これを一般化して，残っている参加者がn名である場合を考えると，最大で$\boxed{(8)}$個の試合を同時に行うことができてそれらの試合後の残り参加者数は$\boxed{(9)}$名となる（性質A）。

続いて上記の考察を元に，大会終了までにかかる所要時間の最小値$L(N)$を求めたい。$L(N)$の単位は「時間」とする。但し，ここでの所要時間は実際に試合を行っている時間のみを考えることとする（移動や休憩の時間は含めない）。例えば参加者が2名であれば1個の試合で優勝者が定まり，1試合にかかる時間は1時間であるので$L(2)=1$となる。参加者が3名の場合について考えると優勝者を定めるまで2個の試合が必要である。但し1個目の試合で勝った参加者が次の試合の参加者となるため，2個の試合を同時に行うことはできない。したがって$L(3)=2$となる。

参加者が8名であれば，所要時間最短の場合の大会は以下のような流れとなるはずである。

1. 最初に第1回戦として$\boxed{(6)}$個の試合を同時に行う。これにより試合後の残り参加者数は$\boxed{(7)}$名となる。
2. 続いて第2回戦として2個の試合を同時に行う。これにより試合後の残り参加者数は2名となる。
3. 最後に決勝戦を行う。

各試合にかかる時間は1時間なので，$L(8) = \boxed{(10)}$ となる。ここで，第1回戦終了後の時点では残り参加者数は $\boxed{(7)}$ 名である。したがって第2回戦以降にかかる所要時間の最小値は $L(\boxed{(11)})$ と表せる。この所要時間と第1回戦にかかる所要時間を合わせたものが $L(8)$ となるはずなので $L(8) = 1 + L(\boxed{(11)})$ と表せる。
参加者が10名の場合について考えると第1回戦で最大限参加者数を減らせば $L(10) = 1 + L(\boxed{(12)})$ となることが分かる。同様にして $L(\boxed{(12)}) = 1 + L(\boxed{(13)})$ となるから，これらを用いれば $L(10) = \boxed{(14)}$ となることが分かる。（$\boxed{(14)}$ は変数や関数を用いない具体的な整数とすること）

関数 $L(n)$ を n の具体的な式で表現する方法を考える。参加者数が増えたときに大会の最短所要時間が減ることはないので $L(n)$ は n に関して広義の単調増加であり

$$n < m \Rightarrow L(n) \le L(m) \quad \text{-- 式(1)}$$

が成り立つ。また，n が2のべき乗で表現できる場合を考えると

$$L(2^k) = 1 + L(2^{k-1}) = \cdots = \boxed{(15)}$$
$$L(2^{k+1}) = \cdots = \boxed{(16)}$$

が成り立つ。（但し，$\boxed{(15)}$ と $\boxed{(16)}$ は変数 k を用いた式で，関数を用いない式で表すこと）
ここで，$n = 2^k + 1$ について $L(n)$ を求めることを考える。前述の性質Aにより，$L(n) = L(2^k + 1) = 1 + L(2^{k-1} + 1)$ が成り立つ。これは k に関する漸化式となっているので，これを用いれば $L(n) = L(2^k + 1) = 1 + L(2^{k-1} + 1) = 2 + L(2^{k-2} + 1) = \cdots = \boxed{(17)}$ となることが分かる（但し，$\boxed{(17)}$ は変数 k を用いた式で，関数を用いない式で表すこと）。
まとめると $L(2^k) = \boxed{(15)}$，$L(2^k + 1) = \boxed{(17)}$，$L(2^{k+1}) = \boxed{(16)}$ となる。したがって，$L(n)$ の単調増加性を考えれば $2^k + 1 < n < 2^{k+1}$ を満たす整数 n について $L(n) = \boxed{(18)}$ と表せる（但し，$\boxed{(18)}$ は変数 k を用いた式で，関数を用いない式で表すこと）。

以上から，整数 n がある整数 k について $2^k < n \le 2^{k+1}$ を満たすとき $L(n) = \boxed{(18)}$ と表せることが分かる。ここで n についての不等式の辺々に，2を底とする対数関数を適用すれば $k < \log_2 n \le k + 1$ が成り立つ。これを用いれば $L(n)$ の値を k を用いない n の関数として表すことができ，具体的には $L(n) = \boxed{(19)}$ のように表せる（但し，$\boxed{(19)}$ は変数 n を用いた式で，変数 k を用いない式で表すこと）。