

日本大学大学院総合基礎科学研究科令和8年度
第2期入学入学試験 [情報科学部門] 解答例

情報科学部門 専門科目 A

```
1. int count(int v, int *s, int *e){
    int cnt = 0;
    for(int *c = s; c <= e; c++){
        if(*c == *s){ cnt += 1; }
    }
    return cnt;
}

int mode(int *s, int *e){
    int mode = *s;
    int cnt = count(*s, s, e);
    for(int *c = s; c <= e; c++){
        int t = count(*c, c, e);
        if(cnt < t){
            mode = *c;
            cnt = t;
        }
    }
    return mode;
}

2. void calc(struct fraction f1, struct fraction f2,
    struct fraction *diff, struct fraction *mult){

    diff->num = f1.num * f2.den - f2.num * f1.den;
    diff->den = f1.den * f2.den;

    mult->num = f1.num * f2.num;
    mult->den = f1.den * f2.den;
}

int main(void){
    struct fraction f1, f2, diff, mult;
    scanf("%d/%d, %d/%d", &f1.num, &f1.den, &f2.num, &f2.den);

    calc(f1, f2, &diff, &mult);
```

```

    printf("%d/%d\n", diff.num, diff.den);
    printf("%d/%d\n", mult.num, mult.den);
}

```

3.

```

(a) void print_reverse( struct cell *list ){
    if(list == NULL){ return ; }
    print_reverse( list->next );
    printf("%d ", list->value);
}

```

```

(b) struct cell *remove_less_than_n(struct cell *list, int n){
    while(list != NULL && list->value < n){
        list = list->next;
    }

```

```

    struct cell *cur = list;
    while( cur-> next != NULL ){
        if(cur->next->value < n){
            cur->next = cur->next->next;
        }
        else {
            cur = cur->next;
        }
    }
    return list;
}

```

4.

```

(a) int n_leaves(struct node *n){
    if(n == NULL){ return 0; }
    int cnt = (n->first_child == NULL? 1: n_leaves(n->first_child));
    return cnt + n_leaves(n->next_sibling);
}

```

```

(b) struct data {
    struct node *node;
    int child_count;
};

```

```

int n_children(struct node *n){

```

```

int count = 0;
for(struct node *child = n->first_child; child != NULL; child = child->next_sibling){
    count += 1;
}
return count;
}

struct data max_children_(struct node *n){
    struct data max;
    max.node = n;
    max.child_count = n_children(n);

    for(struct node *child = n->first_child; child != NULL; child = child->next_sibling){
        struct data tmp = max_children_(child);
        if( max.child_count < tmp.child_count ){
            max.child_count = tmp.child_count;
            max.node = tmp.node;
        }
    }
    return max;
}

struct node *max_children(struct node *n){
    struct data max = max_children_(n);
    return max.node;
}

```

情報科学部門 専門科目 B 1

1. 母平均 μ を持つ母集団からの無作為標本を X_1, \dots, X_n とする。

$$E[\bar{X}] = E\left[\frac{1}{n} \sum_{i=1}^n X_i\right] = \frac{1}{n} \sum_{i=1}^n E[X_i] = \frac{1}{n} n E[X_i] = E[X_i] = \mu$$

2. $a(x_i)$ は、データ点 x_i が属するクラスタ C において、 x_i と同一クラスタ内の他のデータ点との平均非類似度を表す。同一クラスタ C 内での非類似度であるため、この値が小さいほど、クラスタ C が互いに類似したデータ点から構成されていることを意味する。

一方、 $b(x_i)$ は、 x_i が属していない各クラスタ C' に対して計算される、 x_i とクラスタ C' 内のデータ点との平均非類似度の最小値を表す。これは異なるクラスタとの非類似度を表す量であり、値が大きいほど x_i が他のクラスタから良く分離されていることを表す。

以上より、シルエットスコアの分子 $b(x_i) - a(x_i)$ は、 x_i が同一クラスタ内のデータ点とどの程度近く、かつ他クラスタからどの程度離れているかを表す指標である。また分母は、スコアの値を $-1 \sim +1$ の範囲に正規化する役割を果たしている。

このようにシルエットスコアは、クラスタ内の凝集性とクラスタ間の分離性の両方の観点から、クラスタリング結果の妥当性を評価する指標である。

3. 「枝刈り」は、決定木学習における過学習を防ぐことを目的として行われる。事前枝刈りは、分割ノードを選択する段階で適用される枝刈りであり、ノードに含まれるデータ数が少ない場合や、十分に良い分割属性が得られない場合に、それ以上の木の分割を行わずに学習を停止する手法である。

一方、事後枝刈りは、一旦決定木を最後まで構築した後に適用される枝刈りである。例えば誤り削減枝刈りでは、ある部分木をそのまま用いた場合の予測精度と、その部分木を葉ノードに置き換え、検証用データにおけるマジョリティクラスで予測した場合の精度とを比較し、後者の方が高い場合に枝刈りを行う。このようにして、木の簡略化と汎化性能の向上を同時に図ることができる。

4. 条件は「 P を真に包含する任意のパターン Q は、 P よりも小さな支持度を持つ」ことを意味している。すなわち、 P は同一支持度を持つパターン集合において、包含関係の観点から極大なパターンであることを表している。

飽和パターンは、同一の支持度をもつ複数の頻出パターンを代表するものとみなすことができ、単一のパターンでより多くの情報を保持していると考えられる。また、支持度付きの飽和パターン集合から、すべての頻出パターンとその支持度を復元することが可能であるため、飽和パターンは頻出パターン集合の圧縮表現として利用できる。

情報科学部門 専門科目 B 2

1.

(a) 画素 (2,2) 周辺の 3×3 領域は

$$\begin{bmatrix} 10 & 10 & 10 \\ 10 & 50 & 50 \\ 10 & 50 & 50 \end{bmatrix}$$

である。Sobel フィルタ (x 方向) との畳み込みは

$$(-1 \cdot 10 + 0 \cdot 10 + 1 \cdot 10) + (-2 \cdot 10 + 0 \cdot 50 + 2 \cdot 50) + (-1 \cdot 10 + 0 \cdot 50 + 1 \cdot 50) = 120$$

となる。

(b) エッジ検出は高周波成分を強調する処理であるため、雑音も同時に強調されやすい。事前に平滑化を行うことで雑音由来の高周波成分を抑制し、誤検出を低減できる。

(c) 低周波成分は画像の大域的な明るさや緩やかな変化に対応し、高周波成分はエッジや細かな模様などの急激な輝度変化に対応する。

2.

(a) $y_n = \sum_{k=0}^{N-1} x_k h_{n-k}$ (添え字が範囲外の h_{n-k} は 0 とみなす)

(b) $F(\mathbf{y}') = F(\mathbf{x}) \odot F(\mathbf{h})$ (\odot : 要素ごとの積)

(c) 継続時間が十分に短い音 (インパルス) をある部屋で鳴らしたときのその部屋の応答

(d) 例: 録音済みの音源に残響を付加し、特定の部屋の音場を再現する。

3.

(a) $\mathbf{h} = \sigma(W\mathbf{x} + \mathbf{b})$

(b) Sigmoid 関数: $\sigma(x) = \frac{1}{1 + e^{-x}}$
ReLU 関数: $\sigma(x) = \max(0, x)$

勾配消失が起こりやすい活性化関数: Sigmoid 関数

(c) 誤差逆伝播では勾配が層ごとに連鎖律により掛け合わされるため、勾配が小さい関数を用いると深い層ほど勾配が消失する。

4.

(a) 特徴マップの局所領域ごとに代表値を取り出して縮小する処理

(b) 畳み込みを行う際のフィルタの移動幅

情報科学部門 専門科目 B 3

1. 利点 子どもは大人と比べてロボットへの関心を強く持ちやすく、ロボットを意図スタンスで捉えやすいため、積極的に質問を引き出せる可能性が高い。また、児童をセンシングし発話予備動作などを検出することで、質問したい児童への働きかけも可能である。

欠点 ロボットの応答が児童の期待値を下回り、負の適応ギャップが生じたとき、ロボットを叩く、邪魔するといったロボットいじめに発展する可能性がある。ロボットいじめは子どもとのインタラクションで多く報告されている。
2. 環境の相違点 一般的な小学校と大学では、ロボットの動作する環境が大きく異なる。小学校の教室は大学の教室と比べて狭いことが多い。また小学校の教室は画一的な場合が多いが、大学の教室は広さや形状が多様な場合が多い。

対象の相違点 大学生は小学生と比べてロボットに対する警戒心が高く、ロボットとコミュニケーションをとっている姿を他者から見られる恥ずかしさを強く感じる可能性が高い。したがって積極的なコミュニケーションが行われる可能性が相対的に低い。

時間の相違点 小学校では児童は多くの場合時間を自分の教室で過ごす。そのためロボットが教室に導入された場合は長時間・長期間の関係構築ができる。大学では授業ごとに教室が異なるため、ロボットとの関係構築は授業内の限られた時間で行われる。
3. 授業を教員との掛け合いで進める教育用ロボットを提案する。事前に授業資料を読み込むことで想定された授業進行を支援しつつ、教員が一方向的に説明を行っていた授業にメリハリを持たせる。また学生との掛け合いは教員が主導することによって、学生のロボットに対する抵抗感を弱め、学生の質問に対しては「今のはいい質問でしたね」などといって後から介入することで、質問者の学生の満足度の向上に貢献する。ロボットの見た目は人型で、滞りなく対話ができる印象を持たせる。このデザインは、大掛かりなセンシングデバイスも不要であり、教室の大きさや受講者の人数に依存しない。さらにロボットとの深い関係構築ができていなくても有効である。