

令和 8 年度  
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

## 入 学 試 験 問 題

No. 1

コース等	コンピュータ理工学 コース	試 験 科 目	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	---------	-----------------------------

試験時間は 1 時間 30 分です。試験監督から指示があるまで、この表紙をめくってはいけません。次ページ以降に問題が、右上に番号付けされた用紙に分けて出題されています。配点は 140 点です。解答にあたっては、解答用紙の表紙の指示に従いなさい。

解答開始の合図の後、各試験科目の問題について下表中に示す No. の用紙が綴じ込まれていることを確認しなさい。用紙に乱丁・落丁がある場合には、手を挙げて試験監督に知らせなさい。

	試験科目	問題用紙の ページ番号
必須	アルゴリズムとデータ構造, 並びにプログラミング	No. 2~8

解答は、原則、1 問につき 1 枚を使用する。もしも解答用紙のスペースが不足の場合には、手を挙げて試験監督に知らせること。

すべての解答用紙について、受験番号欄に受験番号を記入の上で試験終了後に提出しなさい。本用紙を含むすべての問題用紙についても、回収します。

令和 8 年度  
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

## 入 学 試 験 問 題

No. 2

コース等	コンピュータ理工学 コース	試 験 科 目	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	---------	-----------------------------

問1 以下の問いに答えよ。

- (1) 整列アルゴリズムの安定性とは何か説明せよ。
- (2) 空のスタックに A, B, C, D の順で push する. 4つの文字を push する間, 任意のタイミングで pop できるとする. スタックから pop した際にだけ文字が出力される場合, 以下の選択肢の中でスタックから出力できない文字の列を1つ選択せよ.  
  - (a) BACD
  - (b) BADC
  - (c) BCAD
  - (d) BCDA
  - (e) BDAC
  - (f) BDCA
- (3) ソート済み配列[2, 5, 8, 12, 16, 19, 23, 38, 45, 51, 60, 67, 78, 89, 95]に対して値 16 を二分探索で探す際の比較回数を答えよ。
- (4) オブジェクト指向プログラミングにおける継承 (inheritance)について, 具体例を挙げながら 200 字程度で説明せよ. 説明には継承の利点を含めること。

令和 8 年度  
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

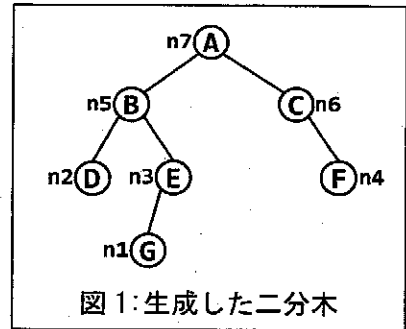
入 学 試 験 問 題

No. 3

コース等	コンピュータ理工学 コース	試験科目	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	------	-----------------------------

問2 以下のコードは二分木を C++言語で実装した例である。

```
template <class T> class BinNode {
private:
    T data; //値を保持するためのメンバ変数
    BinNode<T> *left, *right; //子ノードの番地を保持するためのポインタ 2つ
public:
    BinNode(T d, BinNode<T> *l=NULL, BinNode<T> *r=NULL) {
        data = d;
        left = l; right = r;
    }
};
```



このとき以下の問いに答えよ。

- (1) 以下のコードを実行して図 1 に示す二分木を生成した。  
空白部分のコードを記述せよ。

```
int main(void) {
    BinNode<char> n1('G'), n2('D'), n3('E', &n1), n4('F');
    BinNode<char> n5('B', &n2, &n3), n6('C', NULL, &n4), n7( );
    return 0;
}
```

- (2) 上記で構築した二分木に対して下のコードを実行して n7 から巡回した際の出力結果を示せ。

```
void BinNode::traversePreOrder() { //再帰呼び出しによる深さ優先での巡回
    cout << data << ' '; //自分のノードの値を最初に出力
    if (left != NULL) left ->traversePreOrder(); //左の子をたどる
    if (right != NULL) right->traversePreOrder(); //右の子をたどる
}

n7->traversePreOrder();
```

- (3) 次にノードの巡回順と処理順を変えて、n7 から巡回したところ以下の出力を得た。

F C G E D B A

このような出力となるようなコードを記述せよ。

```
void BinNode::traverseAnotherOrder() { //再帰呼び出しによる深さ優先での巡回
    [ ]
}
```

- (4) 幅優先で巡回した場合の出力はどうか示せ。ただし同じ深さの子ノードは左からたどる。

- (5) キューを使って幅優先で巡回してノードの値を出力するコードを記述せよ。

令和 8 年度  
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

## 入 学 試 験 問 題

No. 4

コース等	コンピュータ理工学 コース	試 験 科 目	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	---------	-----------------------------

問 3 問題用紙 No. 5 から No. 7 に渡って示すソースコードには、とある植物園へ入場するひとつの団体の情報（人数および各人の年齢範囲とこれまでの来園回数）の入力を受け付けたうえで、その中の割引対象者を出力するプログラムが記述されている（C++言語）。この中で、Visitor クラスは、入場者各人を表すクラスであり、そのメンバ変数の age はその入場者の年齢範囲（Child（小人）、Adult（大人）、Senior（シニア）のいずれか）を、また、times はこれまでの来園回数を表す。割引対象者となる条件は、以下のとおりである。

## 割引対象者となる条件

- 団体内に小人が 2 人以上いるうえで、その中ではじめての来園（これまでの来園回数が 0）となる小人は、すべて割引対象者となる。
- 団体内に小人が 1 人以上かつシニアが 1 人以上いるうえで、その中でこれまでの来園回数が 2 以上となるシニアは、すべて割引対象者となる。

このコードについて、以下の問いに答えよ。

- (1) main() 関数の /\*ウ\*/ では、visitor\_factory を用いて、age に対応する Visitor サブクラスのオブジェクトを構築している。この visitor\_factory は /\*イ\*/ の直後で定義されているが、visitor\_factory で参照している create\_child などの定義が欠けている。そこで、create\_child の定義を /\*イ\*/ の位置に記述せよ（create\_adult と create\_senior の定義も必要だが、それらは書かなくてよい）。
- (2) main() 関数は Visitor 型の二次元配列 v を動的に確保しているが、解放処理を行っていない。すべてのオブジェクトを破棄してメモリを解放するコードを /\*オ\*/ の位置に記述せよ。
- (3) このソースファイルをコンパイル・実行し、標準入力に対して下記の下線部のように入力する。このときの出力結果を示せ。なお、check\_discount(...)//\*エ\*/ の行で発生する出力結果のみでよい。

```

団体の入場人数を入力> 4
1人目 1:Child, 2:Adult, 3:Senior> 3
.....これまでの来園回数> 2
2人目 1:Child, 2:Adult, 3:Senior> 2
.....これまでの来園回数> 2
3人目 1:Child, 2:Adult, 3:Senior> 1
.....これまでの来園回数> 0
4人目 1:Child, 2:Adult, 3:Senior> 1
.....これまでの来園回数> 0

```

- (4) Visitor クラス定義の /\*ア\*/ の行にある virtual を削除してコンパイル・実行すると、プログラムの挙動はどのように変化するか。30 字程度で説明せよ。

令和 8 年度  
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

## 入 学 試 験 問 題

No. 5

コース等	コンピュータ理工学 コース	試 験 科 目	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	---------	-----------------------------

```

#include <iostream>
using namespace std;

enum class Age{ Child = 0, Adult, Senior, NumTypes };

class Visitor {
    Age age;
    int times;
public:
    static int num_of[static_cast<int>(Age::NumTypes)];
    const static char *age_string[static_cast<int>(Age::NumTypes)];

    Age get_age() const { return age; }
    void set_age(Age a) {
        age = a;
        ++num_of[static_cast<int>(a)];
    }
    int get_times() const { return times; }
    void set_times(int t) { times = t; }
    virtual bool discount() const { return false; }
};

int Visitor::num_of[static_cast<int>(Age::NumTypes)] = {};
const char *Visitor::age_string[static_cast<int>(Age::NumTypes)] = {
    "Child", "Adult", "Senior" };

class ChildVisitor : public Visitor {
public:
    ChildVisitor() { set_age(Age::Child); }
    bool discount() const {
        return Visitor::num_of[static_cast<int>(Age::Child)] >= 2 &&
            get_age() == Age::Child && get_times() == 0;
    }
};

class AdultVisitor : public Visitor {
public:
    AdultVisitor() { set_age(Age::Adult); }
};

// 次ページへ続く

```

令和 8 年度  
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

## 入 学 試 験 問 題

No. 6

コース等	コンピュータ理工学 コース	試験科目	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	------	-----------------------------

// 前ページからの続き

```
class SeniorVisitor : public Visitor {
public:
    SeniorVisitor() { set_age(Age::Senior); }
    bool discount() const {
        return Visitor::num_of[static_cast<int>(Age::Child )] >= 1 &&
            Visitor::num_of[static_cast<int>(Age::Senior)] >= 1 &&
            get_age() == Age::Senior && get_times() >= 2;
    }
};
```

/\*I\*/

```
Visitor* (*visitor_factory[])() = {
    create_child,
    create_adult,
    create_senior
};
```

```
void check_discount(Visitor* const* v, const int n) {
    for (int i = 0; i < n; ++i)
        if (v[i]->discount())
            cout << i + 1 << "人目 ("
                << Visitor::age_string[static_cast<int>(v[i]->get_age())]
                << ", これまでの来園回数" << v[i]->get_times() << "回)\n";
}
```

// 次ページへ続く



令和 8 年度  
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No. 8

コース等	コンピュータ理工学 コース	試験科目	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	------	-----------------------------

問4 1 つの袋と  $n$  個の荷物が与えられ, 荷物  $i$  ( $i = 1, \dots, n$ ) の重さと価値はそれぞれ  $w_i (> 0)$ ,  $p_i (> 0)$  とする. また, 袋に詰める荷物の重さの上限は  $C (> 0)$  とする. ただし,  $0 < w_i < C$  ( $i = 1, \dots, n$ ) である.

$n$  個の荷物を 1 つの袋に詰め込むナップザック問題を考える. つまり,  $x_i$  を, 荷物  $i$  を袋に詰め込むとき 1, 詰め込まないとき 0 となる変数として, 下記の制約条件を満たす  $x_i$  ( $i = 1, \dots, n$ ) で, 目的関数値が最大となるものを求める.

$$\begin{aligned} \text{目的関数: } & \sum_{i=1}^n p_i x_i \\ \text{制約条件: } & \sum_{i=1}^n w_i x_i \leq C \\ & x_i \in \{0, 1\} \quad (i = 1, \dots, n) \end{aligned}$$

このとき以下の問いに答えよ.

- (1) 荷物の詰め込み方の総数はいくつか.
- (2) この問題を動的計画法で解くとき, 袋に詰め込む荷物の重さの上限を  $u \leq C$ , 詰め込む荷物の候補を  $1 \leq k \leq n$  とする部分問題を考える.

$$\begin{aligned} \text{目的関数: } & \sum_{i=1}^k p_i x_i \\ \text{制約条件: } & \sum_{i=1}^k w_i x_i \leq u \\ & x_i \in \{0, 1\} \quad (i = 1, \dots, k) \end{aligned}$$

この部分問題の最適解 (目的関数の最大値) を表す関数を  $D(k, u)$  とすると,

$$D(1, u) = \begin{cases} 0 & (u < w_1) \\ p_1 & (u \geq w_1) \end{cases}$$

である. 空欄(a), (b)を埋めて,  $D(k, u)$ の値を表す以下の漸化式を完成させよ.

$$D(k, u) = \begin{cases} D(k-1, u) & (u < w_k) \\ \max \{ D(k-1, u), D(k-1, u - \boxed{\text{(a)}}) + \boxed{\text{(b)}} \} & (u \geq w_k) \end{cases}$$

- (3) ナップザック問題を動的計画法で解く場合の時間計算量を  $O$  記法で示せ.
- (4) 次のナップザック問題について, 横方向に荷物の番号  $k$ , 縦方向に重さの上限  $u$  をとった  $D(k, u)$  の二元表を作成し, 最適解を求めよ.

$$\begin{aligned} \text{目的関数: } & x_1 + 4x_2 + 5x_3 + 4x_4 + 2x_5 \\ \text{制約条件: } & x_1 + 3x_2 + 4x_3 + 2x_4 + x_5 \leq 5 \\ & x_i \in \{0, 1\} \quad (i = 1, \dots, 5) \end{aligned}$$