

令和 4 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 分 野 | 専 門 科 目 |
|------|------------------|---------|---------|

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、各分野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。

| | 分野名 | 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. 2~6 |

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 4 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 分 野 | アルゴリズムとデータ構造, 並びにプログラミング |
|------|------------------|---------|-----------------------------|

問 1 以下の設問に答えなさい。

- (1) 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200字程度で説明しなさい。説明には継承の利点を含めること。
- (2) オブジェクト指向プログラミングにおける多態性 (polymorphism) について、具体例を挙げながら200字程度で説明しなさい。説明には多態性を扱うことの利点を含めること。
- (3) 立体を表す抽象的なクラス `Solid` の定義を以下に示す。

```
class Solid {  
public:  
    virtual ~Solid() {}  
    virtual double getSurfaceArea() = 0;  
    virtual double getVolume() = 0;  
};
```

メンバ関数 `getSurfaceArea`, `getVolume` は、それぞれ立体の表面積と体積を返す関数であり、これらの具体的な定義は `Solid` を継承した子クラスで行う。この `Solid` クラスを継承して四角柱を表すクラス `QuadrangularPrism` を定義しなさい。コンストラクタは `QuadrangularPrism(double d, double w, double h)` とし、ここで `d` と `w` は底面の奥行きと幅、`h` は四角柱の高さを表すものとする。

- (4) 立方体クラス `Cube` を定義しなさい。コンストラクタは `Cube(double n)` とし、ここで `n` は立方体の1辺の長さを表す。`Cube` クラスの定義において、親クラスとして `Solid` または `QuadrangularPrism` クラスのどちらを継承して定義すべきかその理由も説明しなさい。

令和 4 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 | 試験分野 | アルゴリズムとデータ構造, 並びにプログラミング |
|------|------------------|------|-----------------------------|

- (5) 複数の Solid オブジェクトの表面積の総和や体積の総和を求めるテンプレート関数 SumUp の定義を以下に示す. この関数は, Solid オブジェクトへのポインタ配列 seq を受け取り, その各要素に対して keySelector が指定するメンバ関数 (このプログラムでは getSurfaceArea() または getVolume()) の返す値の和を求める.

```
template <class KeySelector>
double sumUp(std::vector<Solid*> seq, KeySelector keySelector) {
    double sum = 0;
    for (unsigned int i=0; i < seq.size(); i++)
        sum += keySelector.get(seq[i]);
    return sum;
}
```

```
class SurfaceAreaSelector {
public :
    double get(Solid* s) { return s->getSurfaceArea(); }
};
```

```
class VolumeSelector {
public :
    double get(Solid* s) { return s->getVolume(); }
};
```

この関数を参考にして, 配列中の最大値を持つ要素を取得するテンプレート関数

```
template <class KeySelector>
Solid* findMax(std::vector<Solid*> seq, KeySelector keySelector);
```

を定義しなさい. すなわち findMax は, keySelector で指定したメンバ関数が最大の値を持つ要素を返す関数である. 最大値を持つ要素が複数存在する場合は, インデックスが小さいオブジェクトを返すものとする.

令和4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 | 試験分野 | アルゴリズムとデータ構造 並びにプログラミング |
|------|------------------|------|----------------------------|

問2

| |
|-------------|
| 3 6 2 4 5 1 |
| 3 6 2 4 5 1 |
| 2 3 6 4 5 1 |
| 2 3 4 6 5 1 |
| 2 3 4 5 6 1 |
| 1 2 3 4 5 6 |

図 A

| |
|-------------|
| 3 6 2 4 5 1 |
| 1 6 2 4 5 3 |
| 1 2 6 4 5 3 |
| 1 2 3 4 5 6 |
| 1 2 3 4 5 6 |
| 1 2 3 4 5 6 |

図 B

図 A と図 B は長さ 6 のデータ列 362451 のソーティング過程を示している。以下の設問に答えなさい。

- (1) 図 A と図 B のソーティング法の名称及びそのように判断した理由を回答しなさい。
- (2) 図 A と図 B のソーティング法, どちらがより高速であるか? 最悪, 最良及び平均の計算量について比較しなさい。
- (3) 分割統治法に基づくソーティング法を一つ取り上げ, 上記のデータ列に適用したときの過程を図示しなさい (各ステップで行った処理の説明も加えること)。
- (4) あなたが知っている分割統治法に基づくソーティング法の利点をすべてあげなさい。

令和 4 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

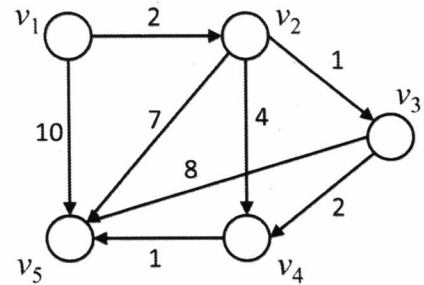
入 学 試 験 問 題

No 5

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

問 3

- (1) 右の有向グラフの隣接行列 W と隣接リスト L を示しなさい。但し隣接行列では、接続されていない頂点間の値は無限大を示す特別な値 ∞ を挿入すること。
- (2) 隣接行列と隣接リストの長所と短所を比較して説明しなさい。
- (3) 以下は出発頂点 s から他の全ての頂点への最短経路を求めるダイクストラ法の擬似コードである。空欄 A と B に挿入すべきコードを答えなさい。



```

1: procedure DIJKSTRA( $V, W, s$ )
   %  $V$ : 有向グラフの頂点集合,  $W$ : 有向グラフの隣接行列,
   %  $s$ : 出発頂点

   % 以下の  $T$  は最短経路が未定の頂点の集合である。また各  $D[v]$  の値は,
   %  $v \in (V - T)$  ならば,  $v$  の最短経路長であり,  $v \in T$  ならば,
   %  $V - T$  の頂点だけを通る  $s$  から  $v$  への最短経路の長さである。

2:    $T \leftarrow (V - \{s\});$ 
3:    $D[s] \leftarrow 0;$ 
4:   for each  $v \in (V - \{s\})$  do
5:      $D[v] \leftarrow W[s, v]$ 
6:   end for
7:   while  $T \neq \phi$  do
8:      $u \leftarrow \arg\text{-min}_{v \in T}(D[v]);$ 
        $\triangleright T$  の頂点  $v$  で  $D[v]$  が最小のものを  $u$  とする
9:      $T \leftarrow (T - \{u\});$ 
10:    for each  $v \in T$  do
11:       $D[v] \leftarrow \min(\text{A}, D[u] + \text{B})$ 
12:    end for
13:  end while
14: end procedure

```

令和 4 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 分 野 | アルゴリズムとデータ構造、 並びにプログラミング |
|------|------------------|---------|-----------------------------|

- (4) 上記の有向グラフに対して出発頂点を v_1 として上の擬似コードを適用する. このときの while ループの実行による T , u と D の変化の様子を以下の表で表す. 空欄の値を記入して表を完成させなさい.

| ループ | T | u | $D[u]$ | $D[v_2]$ | $D[v_3]$ | $D[v_4]$ | $D[v_5]$ |
|-----------|----------------------|-------|--------|----------|----------|----------|----------|
| 実行の 直前 | v_2, v_3, v_4, v_5 | | | 2 | ∞ | ∞ | 10 |
| 1 回目 | | v_2 | | | | | |
| 2 回目 | | | | | | | |
| 3 回目 | | | | | | | |
| 4 回目 | | | | | | | |

- (5) 上記の D の計算では, 各頂点への最短経路の長さは分かる (記録が残る) が, 最短経路そのものは記録されません. 最短経路を記録するためはどう改良すればよいか? について説明しなさい.
- (6) ダイクストラ法が最短経路とその経路長を正しく求めることができる理由を説明しなさい.

令和4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） 後期募集

受験番号

入学試験解答用紙

| | | | |
|------|-----------------------------|----|--|
| コース等 | コンピュータ理工学コース | | |
| 試験分野 | アルゴリズムとデータ構造、 並びにプログラミング | 採点 | |

問3 解答

(1)

(2)

(3)

空欄 A

空欄 B

令和4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） 後期募集

受験番号

入学試験解答用紙

| | | | |
|------|-----------------------------|----|--|
| コース等 | コンピュータ理工学コース | | |
| 試験分野 | アルゴリズムとデータ構造、 並びにプログラミング | 採点 | |

問3 解答

(4)

| ループ | T | u | $D[u]$ | $D[v_2]$ | $D[v_3]$ | $D[v_4]$ | $D[v_5]$ |
|-------|----------------------|-------|--------|----------|----------|----------|----------|
| 実行の直前 | v_2, v_3, v_4, v_5 | | | 2 | ∞ | ∞ | 10 |
| 1回目 | | v_2 | | | | | |
| 2回目 | | | | | | | |
| 3回目 | | | | | | | |
| 4回目 | | | | | | | |

(5)

(6)